EXHIBIT 7

[redacted]

UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION

TQ DELTA, LLC, *Plaintiff,* v. **COMMSCOPE HOLDING COMPANY, INC., COMMSCOPE INC., ARRIS US HOLDINGS, INC., ARRIS SOLUTIONS, INC., ARRIS TECHNOLOGY, INC., and ARRIS ENTERPRISES, LLC** *Defendants*.	CIV. A. NO. 2:21-CV-310-JRG (Lead Case)

TQ DELTA, LLC, *Plaintiff,* v. **NOKIA CORP., NOKIA SOLUTIONS AND NETWORKS OY, and NOKIA OF AMERICA CORP.,** *Defendants*.	CIV. A. NO. 2:21-CV-309-JRG (Member Case)

OPENING EXPERT REPORT OF BRUCE MCNAIR ON THE INVALIDITY OF <u>THE ASSERTED CLAIMS OF THE FAMILY 9 PATENTS</u>

40. My opinions regarding the level of ordinary skill in the art are based on, among others, my experience developing, presenting, and testing novel data communication technologies, my experience teaching and mentoring students at graduate and undergraduate levels, and my experience working in organizations responsible for data transmission standards. Although my qualifications and experience exceed those of a person of ordinary skill in the art, my qualifications enable me to provide opinions regarding the Family 9 patents from the perspective of one of ordinary skill in the art, as I have done in this report.

VII. BACKGROUND OF THE TECHNOLOGY

41. Below, I provide a brief summary of the technology and the state of the art as of the priority date of the Family 9 patents as context for understanding the disclosure and claims of the Family 9 patents and how they compare to the prior art.

42. The Family 9 patents relate to communication systems that utilize interleaving[6] and retransmission for error control. Both interleaving and retransmission were old and well-known techniques by the Family 9 patents' priority date, and communication systems using both interleaving and retransmission were well known and used in commercial products before the priority date of the Family 9 patents. For example, by 1998, the use of both interleaving and retransmission had been standardized for digital subscriber line (DSL) communication systems.

43. Although the claims of the Family 9 patents patent are not limited to DSL, the applicant's presentation of the material in the patents is in the context of DSL. *See, e.g.*, '411 patent, 8:54-67. Therefore, I provide below a discussion of relevant DSL principles and technology.

[6] For convenience, in this report I sometimes use the term "interleaving" to refer to the overall interleaving process, which includes interleaving by the transmitter and deinterleaving performed by the receiver. When necessary, I will distinguish between interleaving and deinterleaving.

44. The *downstream* direction, sometimes abbreviated as "DS," is defined in DSL as the direction from the service provider (or operator) to the subscriber (e.g., from the central office (CO) to the customer's home or office). The *upstream* direction, sometimes abbreviated as "US," is the direction from the subscriber toward the service provider. In ADSL, the transceiver at the subscriber's location, called the "ATU-R," transmits data in the upstream direction and receives data in the downstream direction. Conversely, the ADSL transceiver in the CO (or other location in the service provider's network), called the "ATU-C," transmits data in the downstream direction and receives data in the upstream direction. In VDSL, the transceiver at the subscriber's location, called the "VTU-R," transmits data in the upstream direction and receives data in the downstream direction, and the VDSL transceiver at the service provider's end of the subscriber line, called the "VTU-O," transmits data in the downstream direction and receives data in the upstream direction.

A. <u>Overview of Telecommunications Technologies</u>

45. Telecommunications technologies directed to communicating electronic data (*i.e.*, analog and digital data) have been around for quite some time. Ayre, R., Hinton, K., Gathercole, B., Cornick, K., *A Guide to Broadband Technologies*, 43 Austl. Econ. Rev. 200-08 (June 2010) ("Ayre"), §§1-2.[7] One example is the plain old telephone or telephony service ("POTS"), which was originally created to provide "basic telephone service" between two end users. *See id.* §2. Implementing POTS requires coupling a service provider's central office ("CO") equipment to

[7] As used herein, "communication" and its variations includes: "transmitting" and its variations; "receiving" and its variations; or any combination thereof. *See, e.g.*, *Fundamentals of DSL Technology* (Golden, P., Dedieu, H., Jacobsen, K. eds., Auerbach Publ'ns, 1st ed. 2004, prtg. 2006) ("Jacobsen"), §6.5.5-6 (disclosing that communication requires transmitting a signal, receiving a signal, or a combination of both).

an end user's customer premises equipment via a local loop.[8] *See* Starr, T., Sorbara, M., Cioffi, J., Silverman, P., DSL ADVANCES (Prentice Hall PTR 2003) ("Sorbara"), §1.2, Figure 1.1. A local loop refers to "cables containing twisted pairs of copper wires." Ayre, §2.

46. Early POTS systems could only transport analog data, such as sounds produced by a voice, between a CO and customer premises.[9] *Id.* For example, customer premises equipment may receive analog voice data, convert (*i.e.*, encode) it into analog signals, and transmit the analog signals over or along a local loop to a CO. *Id*. Equipment at the CO can convert (*i.e.*, decode) the received analog signals back into voice data. *Id*.

47. POTS has many shortcomings, such as signal attenuation, crosstalk noise from signals on a wire interfering with other signals on other wires, signal reflections, radio-frequency ("RF") noise, and impulse noise. *Id*. It was also incapable of communicating voice and digital data at the same time because it used a single communication channel comprising a "single narrow frequency band." *See* Ayre, §2.

48. To mitigate some of its shortcomings, POTS was later modified to include modems (e.g., a dial-up or voiceband modem) for communicating analog voice data, other type of analog data, digital data (e.g., video, multimedia, etc.), or a combination thereof. *Id.* A modem generally refers to an apparatus whose functions include modulation, demodulation, and communication of data. *See, e.g.*, Sorbara, §§2.1, 2.1.1-2. A modem is sometimes called a transceiver. *See, e.g.*, Fukushima, 03:36-04:28; Figure 1.

49. There are two types of modulation—analog and digital. *See, e.g.*, FUNDAMENTALS OF DSL TECHNOLOGY (Golden, P., Dedieu, H., Jacobsen, K eds., Auerbach Publ'ns, 1st ed. 2004,

---

[8] CO equipment includes telephone switching equipment, while customer premises equipment includes telephones. *See* Sorbara, §1.2, Figure 1.1.

[9] The term "voice data" and its variations, as used herein in this report, refer to sound. It is to be appreciated that sound and faxes are some of many examples of analog data.

prtg. 2006) ("Jacobsen"), §6.2, Figure 6.1. Demodulation is the reverse of modulation. *Id*.

50. Analog demodulation is directed to mapping a representation of analog or digital data into a symbol, which is a waveform that represents an analog signal.[10] *Id*. The analog data can be represented using one or more of its characteristics, such as amplitude. *Id*. The digital data can be represented using at least one bit. *Id*. Mapping, in this scenario, enables converting the analog or digital data into the analog signal for propagation over a transmission medium. *See, e.g.*, Jacobsen, §6.2.

51. Digital modulation is directed to mapping an analog or digital data representation to a waveform represented by a digital signal. *Id*. Here, mapping allows for converting the analog or digital data into the digital signal for propagation over a transmission medium. *Id*.

52. An integrated services digital network ("ISDN") eventually replaced POTS. *Id*. The ISDN mitigated some of POTS' shortcomings above by enabling faster, synchronous communication of voice and digital data over a local loop. *Id*. Nevertheless, ISDN was not widely adopted because of its cost. *Id*.

53. A "new generation of high-speed data transmission technologies" called digital subscriber line ("xDSL") superseded the ISDN. *Id*. Earlier generations of xDSL replaced the voiceband modem described above with a DSL modem at the customer premises side and a DSL access modem ("DSLAM") at the CO side. *Id*. DSL modems and DSLAMs could use multiple channels comprising "a hundred or more narrow frequency bands simultaneously," as opposed to a single channel formed with a single narrow frequency band. *Id*. Both devices communicate voice data using lower channels formed with lower narrows frequency bands and all other types

[10] Broadly speaking, a waveform is a graphical representation of a signal. *See, e.g.*, Jacobsen, §6.2, Figure 6.1. It can be presented in the time domain, the frequency domain, or both. See, e.g., Sorbara, §2.2.4

of data using higher channels formed with higher narrows frequency bands. *Id*.

54. Implementation details of xDSL are publicly available in specifications created by the ITU-T Study Group 15. *See, e.g.*, ITU-T Recommendation G.995.1, "Overview of digital subscriber line (DSL) Recommendations" (Feb. 2001 ("G.995.1"), §§3.2, 5.1-5.4, 5.8; ITU-T Recommendation G.995.1, "Overview of digital subscriber line (DSL) Recommendations Amendment 1" (Nov. 2001) ("G.995.1-Amendment No. 1"), §5.9. xDSL includes but is not limited to: (i) ADSL1 (ITU-T Recommendation G.992.1, "Asymmetric digital subscriber line (ADSL) transceivers" (July 1999), Sp-ADSL1 (ITU-T Recommendation G.992.2, "Splitterless asymmetric digital subscriber line (ADSL) transceivers" (July 1999)), ADSL2 (ITU-T Recommendation G.992.3, "Asymmetric digital subscriber line transceivers 2 (ADSL2)" (Jan. 2005) ("G.992.3")), Sp-ADSL2 (ITU-T Recommendation G.992.4, "Splitterless asymmetric digital subscriber line transceivers 2 (splitterless ADSL2)" (July 2002)), and ADSL2-plus (ITU-T Recommendation G.992.5, "Asymmetric Digital Subscriber Line (ADSL) transceivers - Extended bandwidth ADSL2 (ADSL2+)" (Jan. 2005) ("G.992.5")); (ii) HDSL (ITU-T Recommendation G.991.1, "High bit rate digital subscriber line (HDSL) transceivers" (Oct. 1998) ("G.991.1")); (iii) SHDSL (ITU-T Recommendation No. G.991.2, "Single-pair high-speed digital subscriber line (SHDSL) transceivers" (Dec. 2003) ("G.991.2"); and (iv) VDSL1 (ITU-T Recommendation G.993.1, "Very high speed digital subscriber line transceivers" (June 2004) ("G.993.1")) and VDSL2 (ITU-T Recommendation G.993.2, "Very high speed digital subscriber line transceivers 2 (VDSL2)" (Feb. 2006) ("G.993.2")).

55. The ADSL family of standards requires asymmetric upstream and downstream

data rates.[11] *See*, *e.g.*, G.992.1, i. HDSL requires "a two-wire bidirectional transceiver for metallic wires" that is directed to transporting "several types of applications" using an echo cancellation method. G.991.1, §1. SHDSL employs symmetric, as opposed to asymmetric, upstream and downstream data rates. G.991.2, i. The VDSL family "permits the transmission of asymmetric and symmetric aggregate data rates" on twisted pairs of copper wires. G.993.1, §1.

B. Open Systems Interconnection Basic Reference Model

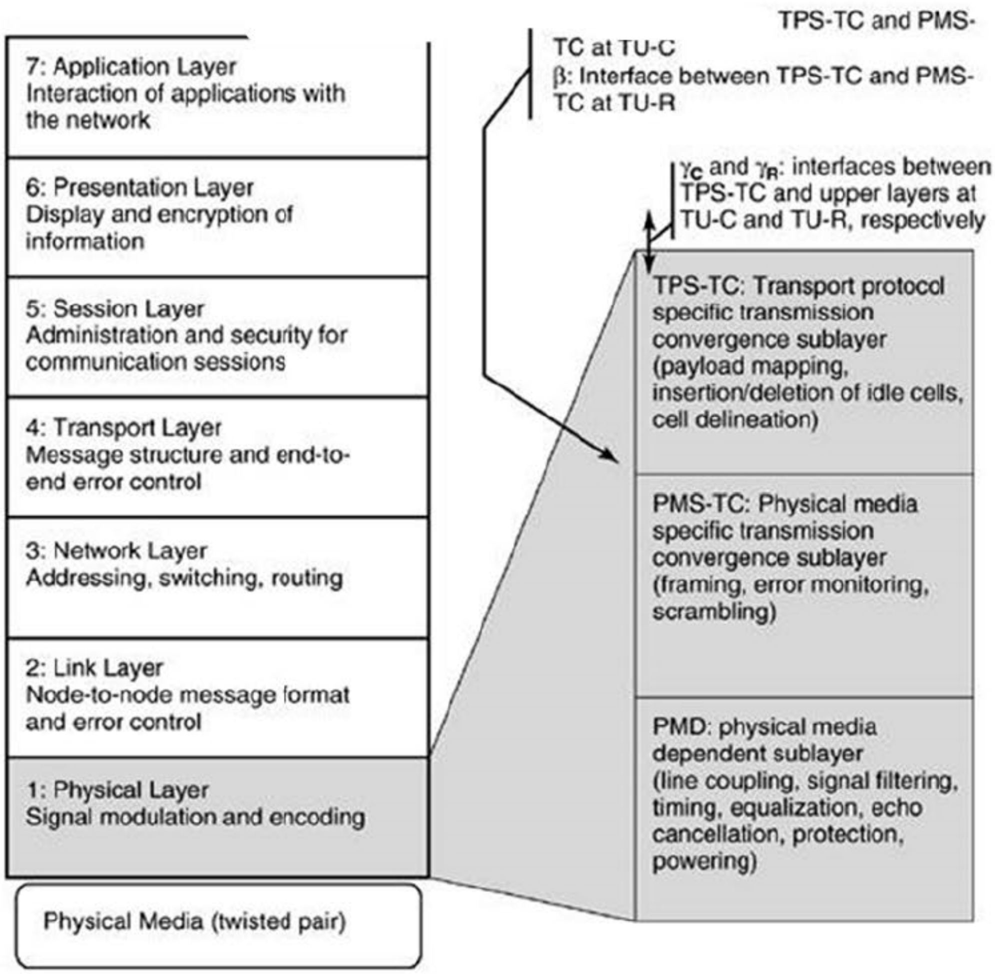
56. Most, if not all, telecommunications technologies using transceivers operate at a physical layer, which is the lowest layer of an open systems interconnection ("OSI") basic reference model (hereinafter "OSI model"). *See e.g.*, Sorbara, §1.4 and Figure 1.6. This model is an "overall framework" that conceptually characterizes and defines data communication in, from, and to an electronic system, such as a telecommunications or computing system. *See* Reynders, §2.4.2, p. 21.

57. In plain English, an OSI model represents an electronic system's information processing and communication capabilities at a high level without regard to the system's complexity, internal structure, or technologies. *See e.g.*, Sorbara, §1.4 and Figure 1.6. OSI models are useful because they allow for different types of electronic systems, including existing and future electronic systems, to be represented in an easy-to-understand and standardized way. *Id*. Every electronic system's OSI model abstracts the system into a maximum of seven layers. *Id.* Each layer includes one or more entities, which are abstract devices, such as programs, functions, or protocols, that implement one or more capabilities of the layer. *See* Reynders, §2.4.2, pp 21-22; *see also* ITU-T Recommendation X.200, "Information technology - Open Systems

[11] Asymmetric data rates require downstream data rates for communications to customer premises equipment to differ from (e.g., exceed) upstream data rates for communications to CO equipment.

Interconnection - Basic Reference Model: The basic model" (July 1994) ("X.200"), §5.2.1.1-5.2.2.7. Furthermore, each layer may or may not be subdivided into sub-layers depending the technologies being represented. *See* X.200, §5.2.2.2, note 2.

58. One example of a seven-layer OSI model is reproduced below with additional detail shown for the physical layer.



See Sorbara, Figure 1.6.

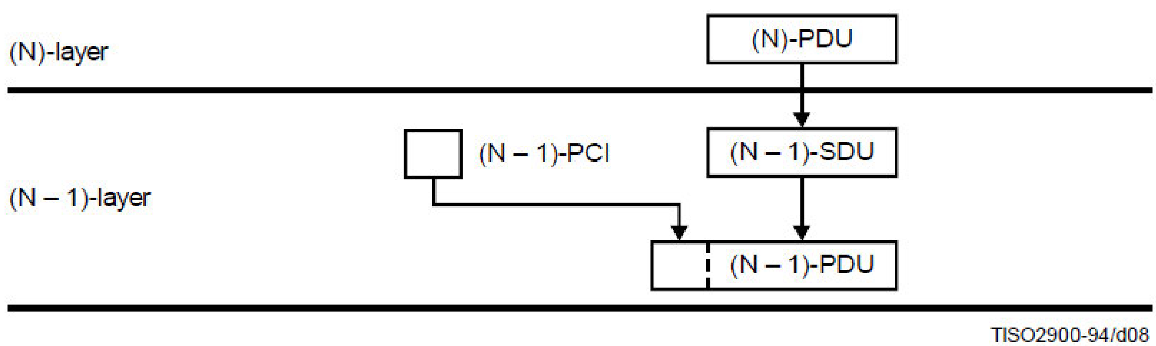
59. In an OSI model, "[i]nformation is transferred in various types of data units." *See* X.200, §5.6.2.1. There are four main types of data units: user data; protocol control information ("PCI"); protocol data unit ("PDU"); and service data unit ("SDU"). *See id.* §§5.6.1.1-5.6.1.5.

60. User data, which is sometimes called payload, refers to substantive information (e.g., voice, video, text, etc.). *Id*. PCI is supplemental information that enables peer layers of communicating systems to coordinate their operation. *Id*. A PDU is generated by a layer using PCI generated by the layer and a data unit received from an adjacent higher layer. *Id*. An SDU is a data unit that is not interpreted by a layer and is received by the layer from an adjacent layer. *Id*.

61. Relationships between the data units are illustrated below.

	Control	Data	Combined
(N)-(N)-peer-entities	(N)-protocol-control-information	(N)-user-data	(N)-protocol-data-unit

See id. Figure 8.



PCI Protocol-control-information
PDU Protocol-data-unit
SDU Service-data-unit

See id. Figure 9.

62. Electronic systems abstracted in OSI models communicate with each other by passing data units through their respective layers and across physical transmission media that communicatively couples the systems. *See id*. Figure 2.9 of Reynders, reproduced below, illustrates two electronic systems in communication using two OSI models. The following Figure

also shows one example of each layer's PDU.

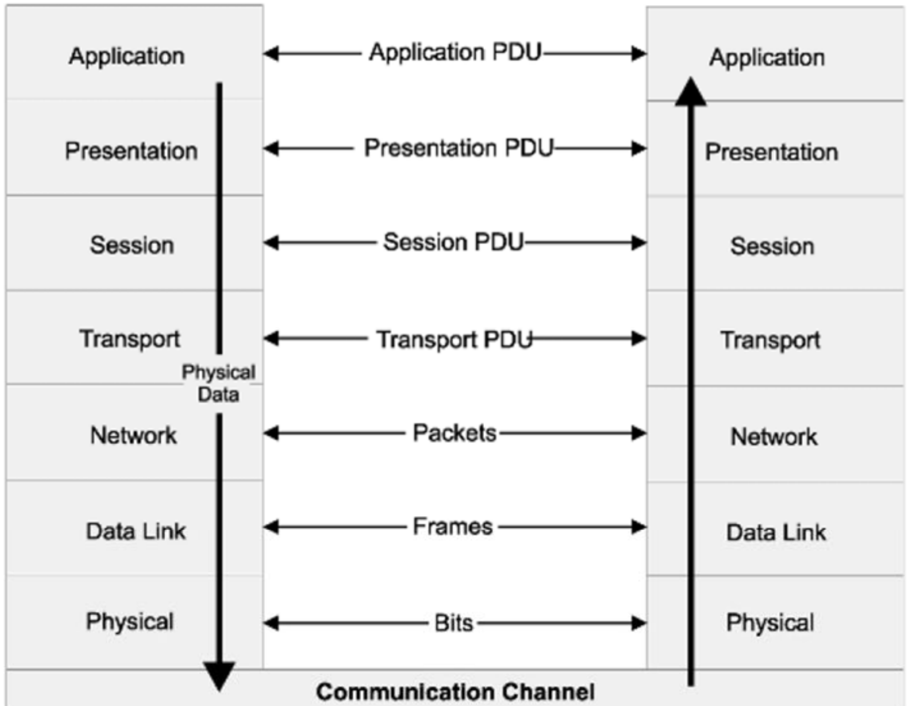


Figure 2.9
Full architecture of OSI model

See Reynders, Figure 2.9, p. 24; *see also id.* §§2.4.1-2.4.10, pp. 20-28.

63. A transmitting system (e.g., the left OSI model in Figure 2.9 above) passes data units "physically down through" its layers, which can include generating a PDU at one of its layers by adding PCI to a data unit received by the layer. *See id*. §2.4.2, pp. 23-24. Furthermore, the receiving system (e.g., the right OSI model in Figure 2.9 above) passes data units physically upward through its layers, which can include generating an SDU at a layer by stripping PCI from a a data unit received by the layer. *Id*.

64. Some electronic systems, such as relay or distribution servers whose main function is passing data between other electronic systems, can be abstracted into OSI models made up of less than seven layers. *See* X.200, §6.1.6. One example is shown below. *See id.* Figure 12.

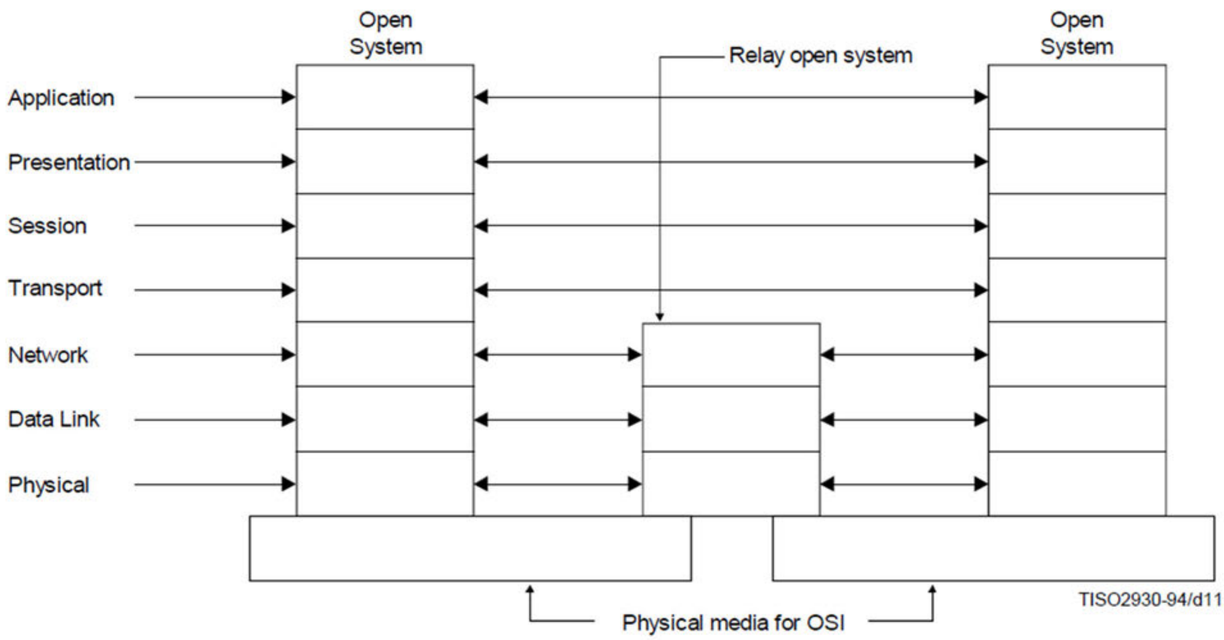


Figure 12 – Communication involving relay open systems

See id.

65. "[P]eer layers across sites communicate with the use of the protocol control information." *See* Reynders, §2.4.2, p. 24. Consequently, each layer of an OSI model "converse[s] with its peer layer at the other end of the communication channel in a virtual ('logical') communication" using PCI. *See id.* p. 21 (quotations in original). This means "there is NO connection or direct communication between the peer layers of" multiple electronic systems that have been abstracted into OSI models. Instead, "all physical communication is across the physical layer, or the lowest layer of the stack." *See id.* p. 24.

66. A PCI data unit can include a header comprising header information, a trailer comprising trailer information, or a combination of both. *See id.* p. 23. Thus, a layer of an OSI model can add a header, a trailer, or both to a data unit received by the layer from an adjacent higher layer. *Id.* The following Figure shows a non-limiting example of a PDU in each layer of a seven-layer OSI model. In the Figure, each PDU includes a non-limiting example of a PCI data

unit.[12]

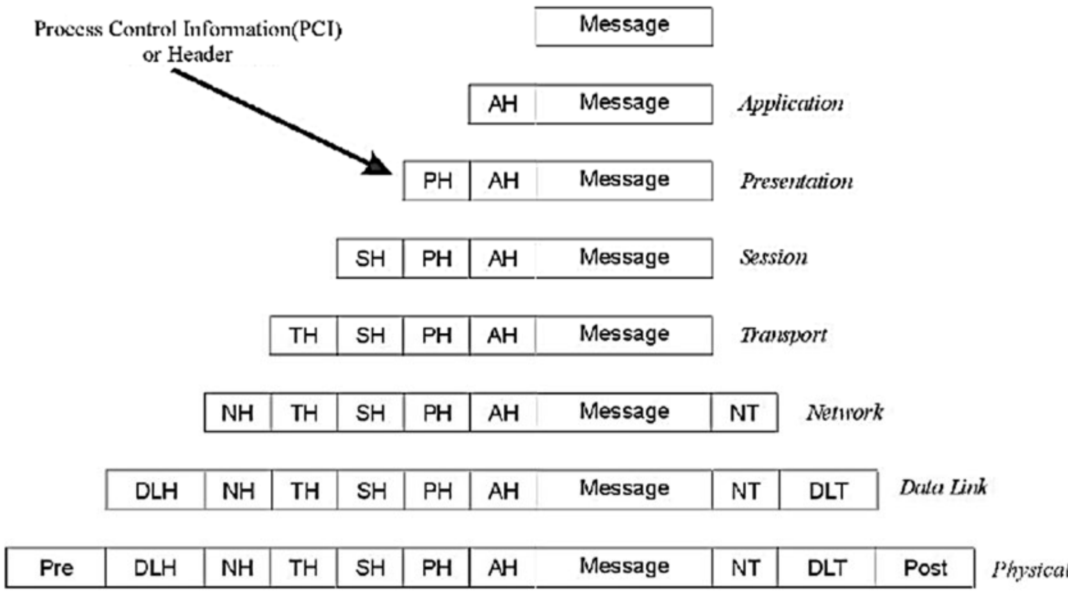


Figure 2.10
OSI message passing

See Reynders, Figure 2.10, p. 24; *see also id.* §2.4.2, pp. 23-24.

C. Shortcomings of Telecommunications Technologies

67. As described in detail above in §VII.B, telecommunication technologies require data communication "across the physical layer." *See id.* §2.4.2, p. 24. One example of this type of telecommunication technology is xDSL, which can be used to exchange analog and/or digital data between multiple devices, such as between xDSL transceivers of a customer premises and a CO. G.995.1, §5.1. Implementing xDSL requires at least one pair of twisted telephone wires to communicatively couple multiple xDSL transceivers, such as xDSL transceivers at a customer premises and a CO. *Id.* The performance, stability, and data rate or reach of the telephone wires

[12] "AH" denotes the header of the application layer's PDU. *See* Reynders, Figure 2.10, p. 24. "PH" denotes the header of the presentation layer's PDU. *Id.* "SH" denotes the header of the session layer's PDU. *Id.* "TH" denotes the header of the transport layer's PDU. *Id.* "NH" and "NT" respectively denote the header and the trailer of the network layer's PDU. *Id.* "DLH" and "DLT" respectively denote the header and the trailer of the data link layer's PDU. *Id.* "Pre" and "Post" respectively denote the header and trailer of the physical layer's PDU. *Id.*

can fluctuate in response to unpredictable "environmental conditions." *See* Jacobsen, §9.1. For example, variations in weather, material quality and age, and electromagnetic energy can introduce noise into communication channels or communicated information, which results in noise that corrupts communicated data with one or more errors. *See id.*

68. Noise can have a negative impact on the quality and stability of telecommunication systems, such as xDSL. Noise can be categorized based on the location of corruption. The categories include, but are not limited to, noise that corrupts data: (i) before a transmitter transmits the data ("hereinafter transmitter noise"); (ii) as the data is passing through the communication channel (hereinafter "channel noise"); and (iii) after a receiver receives the data (hereinafter "receiver noise").

69. Transmitter and receiver noise typically originates from a faulty component in the transmitter and receiver, respectively. *See* Jacobsen, §7.3.2 (disclosing that power requirements of a DMT transceiver may "require components with higher precision" and a failure to design the transceiver with suitable components will result in unwanted transmitter and/or receiver noise).

70. One example of transmitter and receiver noise corrupts input signals processed by one or more components (e.g., an analog filter, an analog front end). *See id*. §§13.1.1, 13.3.2. This type of noise can be attributable to a component's age, material quality, and the like. Another example of transmitter and receiver noise is "quantization noise," which can be introduced into a signal during a quantization operation performed by a digital-to-analog converter ("DAC") of a transmitter or an analog-to-digital converter ("ADC") of a receiver. *Id.* Yet another type is "clipping" which is detrimental to DMT symbols in VDSL2 and caused by a DAC's or ADC's inability to handle a signal beyond its capabilities. *Id*.

71. Channel noise includes crosstalk noise, impulse noise, radio noise, background

noise, or any combination thereof. *See, e.g.*, Sorbara, §2.6. In the three noise categories, crosstalk noise can have the most detrimental effect on the performance of telecommunication systems, such as xDSL. *See* Starr, T., Cioffi, J., Silverman, P., UNDERSTANDING DIGITAL SUBSCRIBER LINE TECHNOLOGY (Prentice Hall PTR 1999) ("Starr"), §3.6.1. Crosstalk noise occurs when an electromagnetic field radiating from each wire in a twisted pair that is part of a cable comprising multiple twisted pairs of wires causes interferes with the signals communicated by the neighboring twisted pairs of wires. *Id.* For example, a signal being communicated by a twisted pair of wires "crosses" into a neighboring twisted pair of wires, where the signal is seen as noise. *Id.* The crosstalk signal can cause a decrease in the signal-to-noise ("SNR") of the transmission on the affected twisted pair of wires. *Id.*

72. There are two types of crosstalk noise: near end crosstalk noise ("NEXT"); and far end crosstalk noise ("FEXT"). *Id*. NEXT occurs when signals are communicated in "opposite directions on two twisted pairs of wires" or "between a transmitter and an 'near-end' receiver." *Id.* (quotations in original). FEXT occurs when signals are communicated "in the same direction on two twisted pairs of wires" or "between a transmitter and a 'far-end' receiver" on the opposite end of a subscriber loop. *Id.* (quotations in original).

73. Impulse noise consists of bursts of energy "inject[ed]" into a twisted pair of wires carrying a signal. *Id.* §3. These bursts are usually temporary, have a high amplitude, and are generated by sources that are external to the twisted pairs of wires. *Id.* These sources include "electromagnetic events" originating from electrical devices such as motors, power lines, voltage sources, and the like. *See id.*

74. Radio noise, which is sometimes referred as or RF interference ("RFI") or radio ingress, can also have negative effects on xDSL communications. *Id.* §3.6.2. Radio noise is caused

by RF signals "imping[ing]" on signals communicated using twisted pairs of wires. *Id.* The RF signals can originate from a wide variety of sources, such as "AM radio broadcasts and amateur (HAM) operator transmissions." *Id.* §3.6.2. Radio noise can reach levels that exceed crosstalk noise levels. *Id.* §3.6.2.

75. Background or intrinsic noise is typically the least detrimental type of noise because it represents a "noise floor"—the lowest possible noise affecting twisted wire pairs. *See* Sorbara, §2.6.1. The noise floor, in one scenario, is based on the assumption that degradation of twisted wire pairs from exposure to thermal energy generated by electronic devices and circuity implementing xDSL prevents twisted pairs from performing optimally. *See id*.

76. Remedying the above and other shortcomings of telecommunications systems, such as xDSL, can assist with improving communication stability and quality.

D. <u>Error Control</u>

77. Data units may become error-corrupted when subjected to noise during processing and communication. Mitigating noise and its effects (i.e., error corruption of data units) is an important aspect of telecommunications technologies. As used herein, the term "error control" refers to noise and error mitigation.

78. xDSL, for example, achieves error control by detecting and correcting noise using several techniques, including cyclic redundancy check ("CRC") coding, forward error correction ("FEC"), interleaving, retransmission, or a combination thereof. *See, e.g.*, Jacobsen, §§9.1-9.6; *see also* Sorbara, Glossary.

1. <u>CRC and FEC Coding</u>

79. xDSL uses CRC codes for error detection. *See* Jacobsen, §9.1. CRC coding appends a CRC code (i.e., one or more additional bits) to a data unit comprised of bits. *Id.* The CRC code is calculated based on the data unit's content (e.g., its bits). *Id.* For example, a CRC

code is a linear, cyclic, block code for appending "*r* symbols of redundancy" to a data with a size of "*k* symbols" to form a "codeword" comprised of "$n = k + r$ symbols." *Id.* §9.2.2 (italics in original). In the resulting CRC codeword, *k* is the number of "unaltered data symbols," *r* is the number of "redundant symbols," and *n* is the total number of symbols that make up the codeword." *Id.* An important capability of CRC error detection is detection of errors *anywhere* in an *n*-symbol CRC codeword—that is, *anywhere* in the *k* unaltered data symbols and *anywhere* in the *r* redundant symbols that make up the *n*-symbol CRC codeword. *See id.* §9.1. The following Figure illustrates an example of CRC coding.

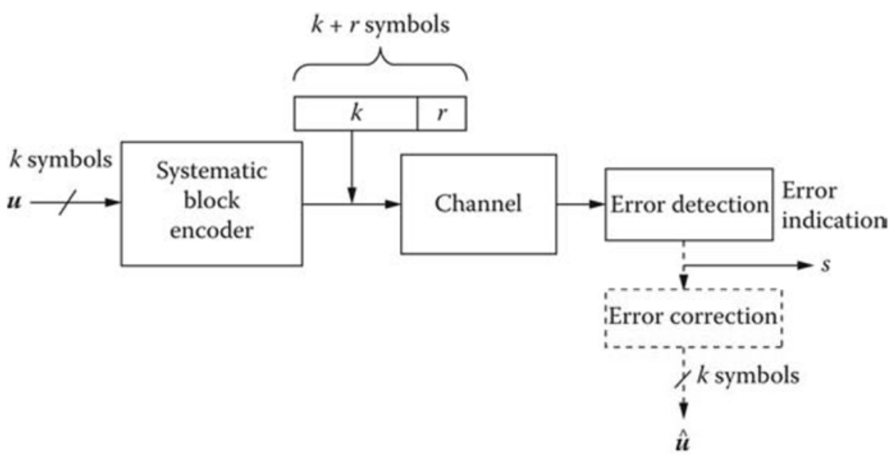


FIGURE 9.1
Illustration of block coding. *k* information symbols are encoded with *r* redundant symbols and transmitted over a channel. At the receiver, errors can be detected or possibly corrected.

Id. Figure 9.1.

80. A transmitter can transmit a generated CRC codeword to a receiver, which can process the CRC codeword to determine whether the original data unit has any errors. *Id.* §9.1. If an error exists, the receiver-side CRC codeword generated at the receiver will differ from the transmitted CRC codeword that originated from the transmitter side. *Id.* When the receiver-and transmitter-side CRC codewords match, it is highly unlikely that any error exists. *Id.* An erroneous CRC codeword can be corrected using one or more error correction techniques. *Id.*

§9.1, Figure 9.1.

81. One error correction technique is FEC, which can detect and "correct errors without retransmitting data." *Id.* §9.2.5. FEC includes using of a coding technique called Hocquenghem, Bose, and Chaudhuri ("BCH") coding. *See id.* §9.2.3. BCH codes can encode a data unit into a BCH codeword capable of error detection and correction. *Id.*

82. The most common type of BCH coding is Reed-Solomon ("RS" or "R-S") coding. *Id.* §9.2.5. RS codes are a subclass of BCH codes that are "used widely in DSL for error correction" because they "provide powerful error correction capability for relatively little overhead." *Id.* A data unit with a size of k unaltered data symbols intended for communication is used, together with an RS code, to generate an RS codeword with a size of n symbols comprised of k number of unaltered data symbols and r number of redundant symbols. *Id.* The RS code enables correction of t errors that occur anywhere in the n-symbol RS codeword at the receiver. *Id.*; *see also* §9.1. In one scenario, an RS code can correct up to $t = (r \div 2)$ errors anywhere in an n-symbol RS codeword made up of k number of unaltered data symbols and r number of redundant symbols. *Id.* §9.2.6. However, RS coding is limited in that, "[i]f the number of errors exceeds the capability of the [RS] code, the errors cannot be corrected." *Id.* §9.2.7. For example, when there are more than t errors in an RS codeword made up of n symbols = (k unaltered data symbols + r redundant symbols), the errors in the RS codeword cannot be corrected by an RS code designed to correct no more than t errors. *Id.* In this and other similar scenarios, other error correction techniques, such as interleaving, retransmission of error-free copies of data, or a combination thereof may be used. Each of these techniques are described in further detail below.

2. <u>Interleaving</u>

83. Interleaving is a type of error control technique that may remedy at least one

limitation of FEC based on RS coding. *Id.* §9.2.7.1. In particular, RS coding can be paired with an interleaving technique at the transmitter to "combat long bursts of errors" in data at the receiver. *Id.* §§9.1, 9.4. An interleaver is "a device that accepts codewords from a finite alphabet and returns the identical codewords but in a different order." *Id.* §9.4. Interleaving is performed after modification of a data unit using an RS code. *Id.* Specifically, interleaving consists of "spread[ing] the data out or shuffl[ing] the data after it is encoded by the Reed-Solomon code. This way, if there is a long burst of errors, the errors will be evenly distributed over many [RS] codewords. If the data is sufficiently shuffled so that each [RS] codeword has only a small number of errors, the [RS] code will correct them." *Id*. Consequently, an interleaver can "spread long strings of errors over several RS codewords." *Id.* The spreading is sufficient to correct a burst of errors if the burst causes no more than t errors in any one RS codeword made up of n symbols = (k unaltered data symbols + r redundant symbols) when the RS code is designed to correct no more than t errors. *Id.*; *see also id.* §9.1.

84. There are at least two types of interleaving—block interleaving and convolutional interleaving. *Id.* Block interleaving requires a block of data represented using RS codewords to be written into columns of a rectangular array and read out from the rows of the rectangular array. *Id.* In this way, a block interleaver generates any entirely new set of RS codewords from the original RS codewords. The number of columns of the rectangular array is a depth of the interleaver, which is a pre-determined separation between adjacent symbols from the same codeword that is measured in symbols. *Id.* Each row of the rectangular array is a newly generated RS codeword that is made up of I number of symbols. *Id.* Note that a new generated RS codeword's size (i.e., I number of symbols) can be less than or equal to the original RS codeword's size (i.e., n number of symbols). *See id.* For example, I can be a divisor of n. *Id.*

Following interleaving, a transmitter communicates the interleaved RS codewords to a receiver, where an inverse of the interleaving process called deinterleaving is applied to the interleaved RS codewords to recover the original RS codeword. *Id.*

85. In some scenarios, a data unit encoded into an RS codewords may be inflicted with eight consecutive errors. Here, each RS codeword has a size n of 7 symbols and an RS code for the RS codeword is designed to correct no more than $t = 2$ errors. In this situation, the RS code would be unable to correct the errors because the RS code cannot correct the extra six errors. The uncorrectable errors could cause decoder errors at the receiver. *See id.* Interleaving may be used to mitigate or eliminate this issue. *See id.* For this example, interleaving will be performed using an interleaver depth d of 4 and an interleaved codeword size of $I = n$, which is 7 symbols. The result is shown below.

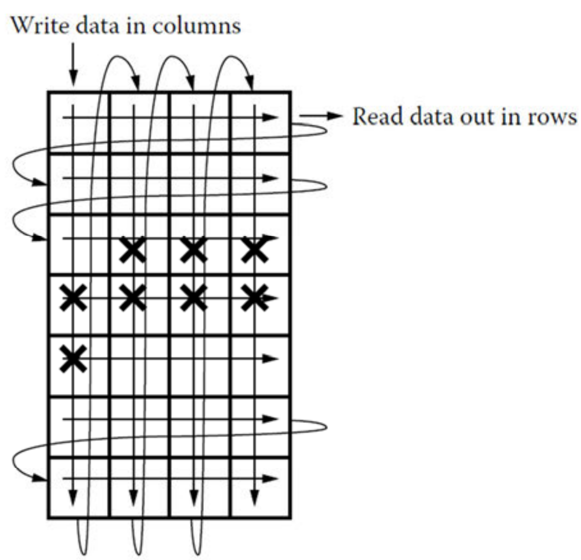


FIGURE 9.12
Block interleaver with seven rows and four columns. Codewords are written in columns and then read out in rows. As an example, each of eight consecutive errors is marked by an X. Each codeword has only two errors.

See id. Figure 9.12.

86. As shown above, interleaving produces four rearranged RS codewords from the original RS codeword, which are read out as rows. In this way, each newly generated RS

codeword only has two errors, which can be fixed by an RS code designed to correct no more than $t = 2$ errors.

87. Block interleaving is not without limitations. For example, a sizeable amount of memory is required to perform block interleaving when compared to using RS coding alone. *Id.* Performing interleaving necessarily introduces latency into telecommunications processes, which would in turn increase data travel times. *See* Bingham, J., ADSL, VDSL, AND MULTICARRIER MODULATION (John Wiley & Sons, Inc. 2000), ("Bingham"), §8.2.5. In block interleaving, all rearranged RS codewords produced by interleaving an original RS codeword must be written into or read out of the interleaver memory before communication can occur. *Id.* Additionally, and in block interleaving, all rearranged RS codewords produced by interleaving an original RS codeword must be written into or read out of the deinterleaver memory before the original RS codeword can be recovered. *Id.* Consequently, as a depth associated with interleaving increases, the number of read/write operations in memory associated with interleaving also increases. As processing times increase, telecommunications are delayed. *Id.*

88. Convolutional interleaving is an alternative to block interleaving that is more memory-efficient. *Id.* This type of interleaving is similar to block interleaving in that symbols are written into a rectangular array in columns and read out in rows. *Id.* Unlike block interleaving, however, convolutional interleaving does not require the memory to be filled before symbols are read. *Id.*; *see also* Jacobsen, §9.4.

3. Retransmission and ARQ

89. Retransmission, commonly referred to as automatic repeat request (ARQ), is an old, well-known, and well-understood technique in which a transmitter resends a block of data, which can be referred to as a packet, to a receiver after the receiver reports, or the transmitter infers, that the received packet was not received error-free. As of 1964, "[e]rror detection

combined with retransmissions on request [had] been used for many years as a means of obtaining reliability in digital data transmission." R.J. Benice and A.H. Frey, Jr., "An Analysis of Retransmission Systems," *IEEE Trans. On Comm. Tech.*, vol. 12, no. 4, 135-145, Dec. 1964 ("Benice I"), 135.

90. Each packet includes overhead in the form of bits (e.g., parity bits or a CRC) that allow the receiver to determine whether the received packet contains errors. If there are no errors, the receiving transceiver sends a positive acknowledgment (ACK) to the transmitting transceiver. If there are errors, the receiving transceiver sends a negative acknowledgment (NACK) to the transmitting transceiver. In response to a NACK, the transmitter sends the packet again. If the transmitting transceiver does not receive either an ACK or NACK within a specified time period after transmitting a packet (referred to in the art as a time-out period), it assumes the packet was not received correctly and retransmits it.

91. The transmitting transceiver retains each transmitted packet in a buffer (memory) until it receives a corresponding ACK. Therefore, implementing a retransmission protocol requires memory in at least the transmitter. The amount of memory needed depends on the retransmission protocol.

92. It was known well before the priority date of the Family 9 patents that retransmission can be used instead of or in addition to FEC coding and/or interleaving. For example, in the 1970s, the use of FEC coding was considered to be both an alternative and a complement to ARQ. *See, e.g.*, Burton, 1300 (disclosing that in high-error-rate environments, "a more symptomatic treatment of the problem would be to imbed a forward-error-correcting code within the ARQ system"); *id.* ("it is likely that the more elegant FEC techniques discussed in this paper will be limited to applications where the customer cannot, or prefers not to, use

retransmission. Where ARQ systems are used, the role of FEC will most likely be a minor one, involving relatively simple short high-rate codes, designed for a limited amount of error correction.").

93. Similarly, ARQ was known to be an alternative to interleaving. *See, e.g.*, ITU-T SG15/Q4 Contribution BI-089, G.gen: ARQ for ADSL Transceivers, Oct. 2000 ("BI-089"), 1 ("the use of ARQ will reduce or eliminate the need for interleaving").

a. ARQ Approaches

94. There are several ARQ approaches. In this expert report, I will describe three of them, which I refer to as "stop-and-wait ARQ," "go-back-*N* ARQ," and "selective repeat ARQ." Each of these approaches has been known in the art since at least the 1980s. *See, e.g.*, Lin *et al.*, "Automatic-Repeat-Request Error-Control Schemes," IEEE Communications Magazine Vol. 22, No. 12 (Dec. 1984) ("Lin") at 6-7.

95. In stop-and-wait ARQ, the transmitter sends a packet and then waits for an ACK before sending the next packet. *See, e.g.*, RFC 3366 (Aug. 2002) at 7. (Depending on the type of communication system, the transmitter might transmit nothing between packets, or it might send "idle" packets or some other signal to help the receiver remain connected.) If the transmitting transceiver receives a NACK, or the time-out period expires without the transmitting receiver having received an ACK, the transmitter retransmits the packet. *See id.*

96. An obvious disadvantage of stop-and-wait ARQ is that it is inefficient because the transmitter must wait for each packet to be positively acknowledged before transmitting the next packet. *See id.* The overall throughput is dependent on whether retransmission is needed, and, if so, how many times.

97. A benefit of stop-and-wait ARQ is that it requires a known, small amount of memory to implement. The transmitter only has to retain the last-transmitted packet while it waits

for a positive acknowledgment. The receiver does not need any additional memory to implement stop-and-wait ARQ because all packets arrive in order, are acknowledged in order, and are passed on in order to the next component or function in the receive path.

98. In contrast to stop-and-wait ARQ, other ARQ protocols—sometimes referred to collectively as "sliding-window protocols"—avoided the need to pause after transmitting each packet by assigning each packet a number, so that multiple packets could be tracked at once. *See, e.g.*, RFC 3366 at 7-8. One example of such an ARQ approach that provides better throughput than stop-and-wait ARQ, but also requires more memory, is referred to as go-back-N ARQ. In this approach, the transmitter sends a packet with index i and then continues to transmit subsequent packets with indices $i + 1$, $i + 2$, etc. while waiting for the ACK for packet i. Assuming that the transmitter sends a total of $N - 1$ additional packets before it expects to receive the ACK or NACK for packet i, when the transmitter receives a NACK for packet i, or it experiences a time-out, it retransmits packet i as well as all of the $N - 1$ packets transmitted after packet i.

99. Relative to stop-and-wait ARQ, go-back-N ARQ provides better throughput because the transmitter does not waste time waiting for ACKs. Go-back-N ARQ is still inefficient, however, because the transmitter retransmits N packets every time one received packet has errors (or is not received). If the communication channel causes a significant number of errors, the throughput of go-back-N ARQ can be poor.

100. In addition, because the transmitter generally has multiple unacknowledged transmitted packets outstanding, each packet needs to include an identifier or index, often referred to as a sequence identifier, so that the transmitter knows which packet is being acknowledged (positively or negatively).

101. Another drawback is that implementing go-back-N ARQ requires memory in the transmitter because the transmitter must temporarily store all N packets it might need to retransmit. The amount of memory needed can be determined and fixed, however: it is the amount needed to store N packets. In operation, assuming the memory size has been selected to accommodate the longest acknowledgment delays and the largest packet sizes, the transmitter's memory requirements will not exceed the amount of memory provided for temporary storage of transmitted packets.

102. A receiver implementing go-back-N ARQ does not need to provide any additional memory for retransmission because the transmitter will resend all packets it sent after the errored packet. Once the receiver detects an errored ith packet, it can simply discard all packets received after the ith packet until it receives the ith packet without errors. Thus, the receiver does not need to provide any additional memory to implement go-back-N ARQ because it will always receive the packets in the correct order.

103. In selective repeat ARQ, the transmitter continues to transmit subsequent packets while waiting for ACKs of previously transmitted packets. When it receives a NACK or experiences a time-out for the ith packet, it retransmits only the ith packet.

104. Relative to stop-and-wait ARQ and go-back-N ARQ, selective repeat provides the best efficiency because the transmitter does not waste time waiting for ACKs, and when it receives a NACK or experiences a time-out for the ith packet, it retransmits only the ith packet.

105. The primary drawback of selective repeat ARQ is its complexity. As in go-back-N ARQ, a transmitter that implements selective repeat ARQ must store all of the packets it might need to retransmit, which includes all packets transmitted but not yet positively acknowledged by the receiver. As compared to the go-back-N approach, the amount of memory needed may need

to be significantly larger if the communication channel is expected to be noisy. If the amount of memory provided turns out to be insufficient in operation due to the channel error rate being high, the transmitter will drop data.

106. Another factor that increases the complexity of selective repeat ARQ is that the transmitter must keep track of which packets have been positively acknowledged and which have not. In addition, the transmitter must insert packets to be retransmitted in its transmit queue.

107. A receiver implementing selective repeat ARQ is also more complex than a receiver implementing either stop-and-wait or go-back-*N* ARQ. First, because the transmitter does not retransmit any packet that was positively acknowledged, the receiver must provide a buffer (memory) to store the packets received correctly after an errored packet so that it can put the received packets back into the correct sequence once it receives the retransmitted packet. As with that of the transmitter, the amount of memory needed by the receiver is difficult to determine because it is dependent on the error rate of the channel. If the amount of memory provided turns out to be insufficient in operation due to the channel error rate being high, the receiver will drop data. *See, e.g.*, Z. Rosberg and N. Shacham, "Resequencing Delay and Buffer Occupancy Under the Selective-Repeat ARQ," *IEEE Trans. On Information Theory*, Vol. 35, No. 1, 166-73, 167, Jan. 1989 ("Rosberg") ("performance under selective-repeat is strongly dependent on the buffer capacity at the receiver. If that capacity does not meet the demand arising from the packet arrivals and the error rate, performance deteriorates drastically. . . . [O]ne has to be very careful when designing the size of the resequencing buffer.").

108. Thus, there are tradeoffs between efficiency and memory requirements of ARQ approaches. Stop-and-wait ARQ requires little memory but also has poor throughput because the transmitter waits for each packet to be positively acknowledged before transmitting the next one.

Go-back-*N* ARQ has better throughput, but it requires more memory at the transmitter. Selective repeat ARQ provides the best throughput, because only packets received in error are retransmitted, but it requires additional memory at both the transmitter and receiver. *See, e.g.*, M.J. Miller and S. Lin, "The Analysis of Some Selective-Repeat ARQ Schemes with Finite Receiver Buffer," *IEEE Trans. On Comm.*, Vol. COM-20, No. 9, 1307-15, Sep. 1981 ("Miller"), 1314 (selective repeat ARQ can achieve "markedly superior throughput performance . . . over Go-Back *N* schemes for channels with high bit rate and large round-trip delay. This improvement is at the expense of required logic complexity and buffer storage provision in the transmitter and receiver."); Rosberg, 166 ("The resequencing buffer requirements [of selective repeat ARQ] and the resulting packet delay constitute major factors in overall system considerations.").

109. By the 1980s, the benefits and drawbacks of the various ARQ approaches in the presence of different types of error environments (i.e., performance in the presence of random errors versus performance in the presence of error bursts) had been studied. It was known, for example, that go-back-*N* ARQ "exhibits much lower message delays than [stop-and-wait ARQ], particularly as the ACK delay increases." D. Towsley, "A Statistical Analysis of ARQ Protocols Operating in a Nonindependent Error Environment," *IEEE Trans. On Comm.*, Vol. COM-29, No. 7, 971-81, Jul. 1981 ("Towsley"), 978. It was also known that go-back-*N* ARQ does not perform as well as selective repeat ARQ "under high error conditions and long ACK delays," but it is "simpler to implement," and the difference in performance "decreases as the error process becomes burstier for a fixed overall error rate." *Id.* Furthermore, it was known that "[a]s the error process becomes burstier, the expected queue lengths under [selective repeat ARQ] always increase." *Id.*

110. Accordingly, it has long been understood that the best ARQ approach to use

depends on the application and the expected error environment. *See, e.g.*, R.J. Benice and A.H. Frey, Jr., "Comparisons of Error Control Techniques," *IEEE Trans. On Comm. Tech.*, vol. 12, no. 4, 146-54, Dec. 1964 ("Benice II"), 149 ("No one of the three logics considered has systems characteristics which make it the most desirable for all applied situations, nor does any one of the logics exhibit universal superiority in efficiency and reliability. Consequently, the selection of one retransmission logic over another for a particular application requires a consideration of the peculiarities of each logic in addition to a determination of which logic offers advantages in efficiency or reliability."); Towsley, 980 ("The throughput of [stop-and-wait ARQ] and [selective repeat ARQ] protocols is determined solely by the overall error rate. In contrast, the throughput of the [go-back-*N* ARQ] protocol is highly sensitive to other aspect of the error process. This sensitivity increases as *N* increases. . . . As the process becomes less random, performance always degrades for the [stop-and-wait ARQ] and [selective repeat ARQ] protocols. This is not always the case for [go-back-*N* ARQ]. We also observed that the performance of [go-back-*N* ARQ] in an environment of bursty errors and high error rates can approach that of [selective repeat ARQ].").

b. <u>ARQ Delay and Delay Jitter</u>

111. As is clear from the above discussion of ARQ approaches, retransmission introduces delay. First, any individual packet that needs to be retransmitted is delayed by at least the sum of (1) the round-trip delay of the channel, (2) the amount of time it takes for the receiving transceiver to determine that the packet has errors and prepare send a NACK, or the time-out period at the transmitter (if the transmitter does not receive an ACK or NACK at all), and (3) the amount of time the transmitting transceiver needs to prepare and retransmit the packet.

112. Second, if a packet that is being retransmitted is part of a sequence of packets that are processed together by the receiver, the need for that packet to be retransmitted delays the

processing of the sequence. For example, if a packet contains a portion of a message, that message cannot be processed by the receiver until all of the portions have been received. If the packets are received out of order, as in selective repeat ARQ, they must be put back into the proper order before subsequent processing. The delay due to the reordering can be referred to as the "resequencing delay." *See, e.g.*, Rosberg, 166 ("Under the selective-repeat ARQ, however, those packets must be stored in the receiver's buffers until they can be sent out in the original order. The buffer needed for this purpose is referred to as a resequencing buffer, and the time that packets spend there as resequencing delay.").

113. It is also clear from the discussion above that unlike the delay caused by interleaving and deinterleaving, which is predictable, bounded (e.g., by a maximum latency constraint), and consistent, the delay caused by retransmission can vary. A packet could experience minimal delay if it is received without errors, or it could experience a large delay if it has to be retransmitted several times. Thus, retransmission results in delay variability, which can also be referred to as delay jitter. Although some applications may be able to tolerate delay jitter (e.g., a file download), others cannot (e.g., telephony). *See, e.g.*, PF-042, §1.1 ("Most real time services, especially interactive applications such as telephony and video teleconferences, can only tolerate a finite amount of latency and jitter. Latency and jitter specifications also affect the memory requirements in the hardware design.").

114. It was known in the DSL field before the priority date of the Family 9 patents that the use of an ARQ protocol results in latency and delay jitter. *See, e.g.*, PF-042, §1.1 (disclosing that using ARQ protocol has a "cost of increasing latency and jitter.").

c. <u>ARQ v. Interleaving</u>

115. It was known before the priority date of the Family 9 patents that ARQ can be used in addition to or instead of interleaving. For example, BI-089, a contribution to the ITU-T

SG15/Q4 group in October of 2000, explained that the use of retransmission "will provide a performance improvement" for DSL systems, and "will reduce or eliminate the need for interleaving." BI-089, 1. BI-089 proposed that "when ARQ is applied, interleaving be reduced or eliminated[,] thereby freeing memory resources for ARQ as well as to reduce or eliminate the interleaver delay so as to compensate for the delay introduced by ARQ." *Id.*, §5.

d. Retransmission in DSL

116. The benefits and drawbacks of ARQ were well known to those working in DSL before the priority date of the Family 9 patents. For example, in October 2000, Catena Networks submitted a contribution to the SG15/Q4 working group that included a proposal to implement data compression in ADSL transceivers. ITU-T SG15/Q4 contribution BI-066, G.gen: G.dmt.bis: G.lite.bis: Strawman Proposal for the Implementation of Data Compression in ADSL Modems (Oct. 2000) ("BI-066"), 1 ("This document provides a strawman proposal for the implementation of data compression as a sub-function of the ATM-TC."). As a starting point, the proposal described how voiceband modems implement data compression, including their use of V.42, which BI-066 describes as "a generic framing protocol, which creates an error free link, based on a less reliable platform below it . . . by adding error correction using an ARQ protocol, which requests a retransmission of data, if the receiver detects a bit error." BI-066, 2. The author of BI-066 considered this approach to have "issues," including "[u]npredictable latency requirements due to V.42 ARQ." *Id.* Another issue was that "[t]he repeated retransmissions introduce jitter and buffering requirements in the ADSL link," which is "undesirable because ADSL has a relatively high ratio of latency to data rate." *Id.* BI-066 noted that "the buffers [for ADSL] have to be larger than those implemented in voice band modems," and that "[i]n the time that it takes a retransmission request to return from the far end, the transmitter has to buffer all the data that it has been transmitting." *Id.* The buffering requirement is exacerbated because "this time is

multiplied by the maximum number of retransmissions allowed." *Id.* BI-066 noted that SG15/Q4 could nevertheless adopt such an approach and "accept the limitations that come with an ARQ protocol." *Id.*, 5.

117. The use of ARQ for ADSL was described by Alcatel in a conference paper in 2000. P. Antoine, "Parallel Concatenated Trellis Coded Modulation with Automatic Repeat Request for ADSL Applications," in *Proc. of 2000 IEEE Intl. Conf. on Comms.* (ICC 2000), Vol. 2, pp. 1075-79 ("Antoine"), 1075. The paper proposed to use ARQ along with parallel concatenated trellis coded modulation (PCTCM) for ADSL. *Id.* Recognizing that "ARQ is a well known method to achieve high reliability in digital transmission schemes," the paper suggested that "ARQ based on PCTCM is particularly suitable for ADSL applications." *Id.*, 1077. Although ARQ could ordinarily cause a substantial decrease in throughput, limiting the number of retransmissions to one would lead to an acceptable reduction in throughput for ADSL because the error rate is "very low," meaning that "very few sequences have to be retransmitted." *Id.*

118. It was well known as of the Family 9 patents' priority date that retransmission can be particularly helpful in the presence of certain types of noise likely to affect DSL, including impulse noise. For example, Alcatel noted in its 2000 conference paper that one benefit of using ARQ in an ADSL system is that "it makes the system immune to impulse noise." *Id.*, 1079; *see also id.*, 1077 ("ARQ based on PCTCM improves robustness against impulse noise.").

119. 3Com made similar observations in its contribution BI-089 to the SG15/Q4 working group, which discusses the benefits of using ARQ along with Reed-Solomon FEC coding in ADSL. BI-089, Abstract. After noting that "the data in ADSL channels occasionally gets corrupted even with adding a 4-6 dB margin," 3Com speculated that the burst errors observed in ADSL "must be" due to impulse noise. *Id.*, §3. Because "[i]t is already known that ARQ offers

substantial performance improvements in bursty channels," (*id.*, §4), "ARQ is a natural and convenient method to fight the impulse noise." *Id.*, §3. 3; *see also id.* ("The real benefit of ARQ is in the bursty channel where the likelihood of FEC failing to correct a frame increases.").

120. Retransmission was not merely known in the DSL field well before the priority date of the Family 9 patents. It had also been incorporated into DSL standards. *See, e.g.*, G.993.2, §12.2.2.1-2 (indicating that a data unit may be retransmitted at least once to improve the likelihood that the data unit is received without errors or in response to the data unit being received with errors). For example, when a DSL transceiver on one end of a line wants to establish a connection with a DSL transceiver on the other end of the line, the transceivers execute what is known as a "handshake" procedure before beginning the initialization procedure specified for the type of DSL (e.g., ADSL2, VDSL2, etc.). The handshake procedures for DSL are standardized in ITU-T Recommendation G.994.1, which is sometimes also referred to as "G.hs" or "G.handshake." By 2003, G.handshake included a "Retransmission Message" that allowed either transceiver to request retransmission "in response to the reception of an errored frame." ITU-T Recommendation G.994.1, Handshake procedures for digital subscriber line (DSL) transceivers, (05/2003) ("G.994.1"), §7.15; *see also id.*, §10.5 ("Retransmission transactions occur whenever an HSTU [handshake transceiver unit] receives an errored frame and wishes to transmit the REQ-RTX [retransmission request] message to initiate a retransmission instead of transmitting the NAK-EF [abort handshake] message."), §4 (indicating that HSTU stands for "Handshake Transceiver Unit"), §12 (indicating that NAK-EF message indicates that receiving transceiver is aborting handshake session).

4. <u>Combinations of Techniques for Error Control</u>

121. Error control can also achieved by combining at least two of the preceding techniques. *See, e.g.*, Fukushima, 26:45-28:34 (indicating that at least two of CRC coding, FEC,

and retransmission may be used together); *see also* U.S. Patent No. 5,907,563 ("Takeuchi"), Figures 7-8 and accompanying text; Lin at 11-15 (discussing "Hybrid ARQ Error-Control Schemes").

E. <u>Memory Management</u>

122. Before the priority date of the Family 9 patents, it was known in the art that memory is a finite resource that must be managed. *See, e.g.*, ITU-T SG15/Q4 contribution PF-042, The proposed MAC for PNT3, ("PF-042") (Aug. 2003), §3.2.1 ("Failure to balance memory and media resource allocation can cause over-utilization of scarce memory resources by aggressive best-effort applications (e.g. FTP). Efficient memory management is also important in designing consumer affordable consumer products, because memory is a costly resource.").

123. As I explain further in the prior art discussion below, the idea of a transceiver sharing a common memory for multiple functions was well known as of the priority date of the Family 9 patents.

124. There are two ways to provide memory for the functions of an apparatus or system. One way is to provide a dedicated, permanently allocated quantity of memory to each function so that each function is always guaranteed access to its assigned memory. I will refer to this approach as using "dedicated memory." One advantage of using dedicated memory is that resource management is simple because each function that needs memory always has access to its assigned memory, and the quantity of memory that is available is always known. The obvious disadvantage of using dedicated memory is that the amount of memory allocated to each function must be the maximum amount of memory that function might need, or there is a risk that the function will not have enough memory. Providing for each function the maximum amount of memory that function might ever need is expensive. It may be the case, however, that under typical conditions, various functions do not need some or all of their assigned memory, in which case the assigned memory

is underutilized.

125. The other way to provide memory for the functions of an apparatus or system is to provide a pool of memory that can be flexibly allocated to different functions at different times depending on need. I will refer to this approach as using "shared memory." One advantage of shared memory is its flexibility. Memory that might otherwise be unused if always dedicated to a particular function can be assigned (e.g., temporarily) to a function that needs it. The disadvantage of using shared memory is that assignments of memory must be managed to avoid conflicts between functions (e.g., if two functions try to use the same memory locations at the same time) and to ensure that the memory allocations to various functions do not exceed the total amount of memory available.

126. By the priority date of the Family 9 patents, the disadvantages of dedicated memory were well known. *See, e.g.*, U.S. 7,266,132 ("Liu"), 1:36-47 ("Static memory used by a network device gives each data for each channel a set amount of memory space for storage. This constant and predetermined memory space size is established conservatively to account for all possible data demands in a particular channel. For static allocation to work, the static allocation must allocate for the worst case. Since the majority of bit streams being processed are much smaller than this conservative estimation, static memory schemes usually result in excessive memory allotment. This increases network device costs.").

127. By the mid-1990s, it was commonplace for computers to use shared memory. *See, e.g.*, J.L. Hennessy and D.A. Patterson, "Computer Architecture A Quantitative Approach, Second Edition," 1996 ("Hennessy"), p. 439 ("It would be too expensive to dedicate a full-address-space worth of memory for each process, especially since many processes use only a small part of their address space. Hence, there must be a means of sharing a smaller amount of

physical memory among many processes. One way to do this, *virtual memory*, divides physical memory into blocks and allocates them to different processes.") (italics in original). For example, the C programing language has a built-in function going back to its 1970s origins that allows a program to dynamically request memory allocation. *See*, *e.g.*, Brian Kernighan, Dennis Richie, The C Programming Language (2nd Ed. Prentice Hall 1988) (1978) at 143. The function to allocate memory is malloc(). *Id.* The function calloc() allocates contiguous blocks while free() releases memory allocations. *Id.* at 167.

F. <u>Shared Interleaver/Deinterleaver Memory</u>

128. The use of shared memory for interleaving and deinterleaving was well known as of the priority date of the Family 9 patents.

129. For convenience, I will sometimes distinguish in this Report between a "near-end" transceiver and the "far-end" transceiver that is on the other side of a communication channel from the near-end transceiver. When the transmitter of a near-end transceiver performs interleaving, the transmitter needs to have access to an amount of memory sufficient for the transmitter to perform the interleaving procedure. Likewise, when the transmitter in the far-end transceiver performs interleaving, the near-end transceiver's receiver must have access to an amount of memory sufficient to enable it to deinterleave the data interleaved by the far-end transceiver's transmitter. And, of course, the far-end transceiver needs sufficient memory to deinterleave the data transmitted by the near-end transmitter and to interleave the data it is transmitting to the near-end receiver.

130. There are two ways to provide the memory needed for a transceiver's interleaving and deinterleaving procedures. The first way is to provide a specified amount of dedicated interleaver memory and a specified amount of dedicated deinterleaver memory. The transmitter has exclusive access to the interleaver memory and can, at least in theory, use as much as all of

the interleaver memory for interleaving. Similarly, the receiver has exclusive access to the deinterleaver memory and can, at least in theory, use as much as all of the deinterleaver memory for deinterleaving.

131. The second way to meet the transceiver's memory requirements for interleaving and deinterleaving is to provide shared memory that the transceiver can partition between its interleaver and deinterleaver. Use of a shared memory for interleaving and deinterleaving, including in DSL, was well known before the priority date of the Family 9 patents. *See, e.g.*, U.S. Patent Pub. No. 2005/0034046 by Berkmann *et al.* ("Berkmann"), §(57) (Abstract) (disclosing a "combined interleaving and deinterleaving circuit" with "data memory (RAM) for temporary storage of the data to be interleaved and deinterleaved"); U.S. Patent No. 6,707,822 to Fadavi-Ardekani *et al.* ("Fadavi-Ardekani"), §(57) (Abstract) (disclosing an "Interleave/De-Interleave Memory" that "is shared by multiple ADSL sessions and by the transmit and receive processes within an individual session"); U.S. Patent No. 6,381,728 to Kang ("Kang"), col. 5:35-38 (disclosing "double buffering [that] allows the channel interleaver memory to be used, along with the app memory, as the turbo deinterleaver memory"); U.S. Patent Application Publication No. 2003/0021338 to Mazzoni *et al.* ("Mazzoni"), §(57) (Abstract) (disclosing memory having "a first memory space ESM1 assigned to the interleaver means and a second memory space ESM2 assigned to the deinterleaving means"); *id.*, ¶[0002] ("The invention is advantageously be applied to a (VDSL) (Very High Rate Digital Subscriber Line) environment or system. . . ."); U.S. Patent No. 5,751,741 to Voith *et al.* ("Voith"), col. 4:47-50 (disclosing external interleave/deinterleave memory used by both transmitter for interleaving and receiver for deinterleaving); *id.* at col. 2:61-64 ("Generally, the present invention provides an ADSL transceiver. . . .").

132. Once the shared memory has been partitioned between the transmitter's interleaver

and the receiver's deinterleaver, the transmitter has exclusive access to the portion of the shared memory allocated for interleaving and can, at least in theory, use as much as all of that portion of the shared memory for interleaving. Similarly, the receiver has exclusive access to the portion of the shared memory allocated for deinterleaving and can, at least in theory, use as much as all of that portion of the shared memory for deinterleaving. When a connection terminates, the shared memory can thereafter be repartitioned, possibly differently, when new connections are established.

133. Because the interleaving procedure is performed by one transceiver, and the corresponding deinterleaving procedure is performed by a different transceiver, it is necessary for at least one of the transceivers to know the capabilities or requirements of the other transceiver in order to configure the interleaver (or deinterleaver). Otherwise, for example, the transceiver performing the interleaving procedure could use an interleave depth that requires more memory than is available to the transceiver performing the deinterleaving procedure.

134. For example, before the near-end transceiver can configure its interleaver, which establishes how much of the memory available for interleaving (in either a dedicated memory or shared memory) the transmitter will use, it needs to know something about the far-end transceiver's deinterleaving capabilities; otherwise, the near-end transmitter could perform an interleaving procedure that the far-end receiver is incapable of reversing (e.g., because it does not have sufficient memory). In addition, before the near-end transceiver can configure its deinterleaver, which establishes how much of the memory available for deinterleaving (in either a dedicated memory or shared memory) the receiver will use, it needs to know how the far-end transceiver will be interleaving the data; otherwise, the near-end receiver will be unable to perform the corresponding deinterleaving procedure.

135. As discussed below, since 1995, the ADSL and VDSL standards have provided for the near-end and far-end transceivers to communicate their interleaving requirements and/or capabilities to each other during an initialization procedure.

G. <u>PTM-TC</u>

136. The acronym "PTM-TC" stands for "Packet Transfer Mode – Transmission Convergence," a DSL-specific term, but the underlying concept is not unique to DSL. "Packet transfer" refers to the fact that data in a DSL system is not transmitted as a continuous data stream but is separated into discrete packets. "Transmission convergence" refers to incoming data: if incoming data is already in packet form, it can be transmitted as-is, but if it is a continuous stream, it will be rearranged and broken into packets.

137. In DSL, information that is to be conveyed across the network may originate in the form of continuous data streams or information that has already been formed into packets. If the information originated as a continuous data stream, it must be repackaged into discrete packets for transmission. On the other hand, if the information was already in packet form, for instance, IP (Internet Protocol) packets, it could be simply be trimmed or padded to fit into the desired DSL packet format, applying the appropriate QoS, error control and sequencing headers. Information that is transmitted using forward error correction is grouped into a series of bits of information and coding redundancy, the total collection referred to as a "codeword."

138. Various DSL standards prior to the priority date required the use of PTM-TC and therefore PTM-TC codewords. *See, e.g.*, G.993.1, 143 ("In the transmit direction… [t]he corresponding TPS-TC (PTM-TC) receives the packet from the γ interface, encapsulates it into a special frame (PTM-TC frame) and maps into the PMS-TC frame (transmission frame) for transmission over the VDSL link"). Indeed, the '055 Patent admits that PTM-TC codewords were known in the art prior to the alleged invention. '055 Patent at 10:25-31 ("For reference,

'Annex A' which is of record in the identified provisional filing and incorporated by reference herein contains the PTM-TC of ADSL2 and VDSL2 systems as specified in the ITU-T G.992.3 ADSL2 standard."); *See* TQD_TX-00001572-1584.

H. <u>Overview of DSL Standards Recommendations</u>

139. Many of the aspects of DSL have been standardized. These aspects include the use of Reed-Solomon coding and interleaving to mitigate errors, and the use of retransmission when a message or message portion is either not received at all or has errors.

140. Every DSL standard defines procedures to enable the transceivers on either end of a subscriber line to establish a connection and procedures that the transceivers conduct to maintain an established connection. The state during which the DSL transceivers establish a connection is referred to as *initialization*, and the state that immediately follows initialization, during which the transceivers can transfer user data, is called *Showtime*.

141. Multiple standardization organizations have been involved in sometimes-overlapping work. This section introduces two standardization organizations whose work predated the Family 9 patents, explains their relationship, and introduces the standards that are particularly relevant to the subject matter of this review and pre-date the priority date of the Family 9 patents.

142. The Alliance for Telecommunication Industry Solutions (ATIS) had a large role in the early standardization of ADSL. ATIS is a standardization organization based in the United States and accredited by the American National Standards Institute (ANSI). Generally, ATIS seeks to develop technical standards that ensure interoperable end-to-end telecommunication solutions that can be timely implemented.

143. ATIS was working to standardize DSL by the early 1990s. The T1E1.4 working group of ATIS carried out the earliest work on ADSL standardization, and, in 1995, ATIS

published the ANSI ADSL standard T1.413, which I will refer to herein as "T1.413 Issue 1." The T1E1.4 working group continued its work on ADSL after 1995, and, in 1998, ATIS published a revision of T1.413 Issue 1, which I will refer to as "T1.413 Issue 2."

144. The other standardization organization relevant to this review is the International Telecommunication Union (ITU). The ITU is an agency of the United Nations that specializes in information and communication technologies. The Telecommunications Sector of the ITU (known as the ITU-T) defines transceiver specifications for use internationally. The ITU-T generates standards called "Recommendations" for all fields of telecommunications.

145. Work in the ITU-T is carried out by Study Groups, and DSL recommendations were and are carried out by Study Group 15. The work of Study Group 15 is partitioned into work areas known as "Questions." DSL recommendation work takes place within Study Group 15, Question 4 (abbreviated herein as SG15/Q4).

146. In 1997, SG15/Q4 began working on ADSL. Among other projects, SG15/Q4 established the projects known as G.dmt and G.lite. At that time, T1.413 Issue 1 was in force, and T1E1.4 was developing what eventually became T1.413 Issue 2. The G.dmt Recommendation was expected essentially to adopt T1.413 Issue 2 and add annexes addressing country-specific issues and requirements. The work in G.dmt was eventually released in 1999 as ITU-T Recommendation G.992.1, much of which is identical to T1.413 Issue 2. The result of the G.lite work was expected to be a lower-speed version of ADSL. The work in G.lite was eventually released, also in 1999, as ITU-T Recommendation G.992.2.

147. Once T1E1.4 completed T1.413 Issue 2, SG15/Q4 became the lead standards working group responsible for defining DSL transceiver recommendations. After releasing T1.413 Issue 2, T1E1.4 focused less on transceiver recommendations and more on providing

inputs to SG15/Q4 regarding North American requirements and preferences for emerging DSL standards.

148. By the Family 9 patents' purported priority date in 2006, the in-force ITU-T Recommendations relevant to ADSL, and relevant to this review, included G.992.1, G.992.2, G.992.3, G.992.4, G.992.5, G.993.1, and G.993.2. Each of these Recommendations specifies techniques for transmitting a range of bit rates over the existing copper local network from high bit rates over relatively short distances to lower bit rates over longer distances. Like T1.413 Issue 1 and T1.413 Issue 2, all of the ITU-T's ADSL Recommendations specify the use of DMT modulation.

149. The Recommendation G.992.1, entitled "Asymmetric Digital Subscriber Line (ADSL) Transceivers" and sometimes referred to by those skilled in the art as "ADSL1" or "G.dmt," specifies the physical layer characteristics of an ADSL transceiver at the subscriber line interface. G.992.1 specifies transceiver requirements for both the transceiver at the telephone company's central office (the ATU-C) and the transceiver at the subscriber's premises (the ATU-R) to enable connections that support at least 6.144 Mbit/s in the downstream direction (i.e., toward the subscriber) and at least 640 kbit/s in the upstream direction (i.e., away from the subscriber) without interfering with plain old telephone signals (POTS) on the same subscriber line.

150. Like G.992.1, Recommendation G.992.2, entitled "Splitterless Asymmetric Digital Subscriber Line (ADSL) Transceivers" and sometimes referred to as "G.lite," specifies the physical layer characteristics of an ADSL transceiver at the subscriber line interface. But unlike G.992.1, which was designed to maximize performance, G.992.2 was designed to provide lower bit rates (i.e., a maximum of 1.536 Mbit/s downstream and 512 kbit/s upstream) by

simplifying various aspects of the transceiver, such as not requiring a splitter to separate the DSL signals from POTS signals on the subscriber line.

151. Recommendation G.992.3, entitled "Asymmetric Digital Subscriber Line Transceivers 2 (ADSL2)" and sometimes referred to as "ADSL2" or, during its development, as "G.dmt.bis," builds on many of the aspects of G.992.1 to enable connections that support at least 8 Mbit/s downstream and at least 800 kbit/s upstream (hence the "Asymmetric" part of its name).

152. Recommendation G.992.4, entitled "Splitterless asymmetric digital subscriber line transceivers 2 (splitterless ADSL2)," referred to as "G.lite.bis" during its development, builds on G.992.2, adding a number of features and capabilities.

153. Another relevant standard in existence on the Family 9 patents' priority date is G.994.1, which I mentioned briefly above (§IV.A.7, *supra*). G.994.1, which is often referred to as "G dot handshake," is entitled "Handshake procedures for digital subscriber line (DSL) transceivers." The purpose of G.994.1 is to provide "a flexible mechanism for digital subscriber line (DSL) transceivers to exchange capabilities and to select a common mode of operation." G.994.1, i. The procedures defined in G.994.1 are "an integral part of the start-up procedure for ITU-T Recs G.991.2, G.992.1, G.992.2, G.992.3, G.992.4 and G.992.5." G.994.1, i.

154. Specifically, G.994.1 "defines signals, messages, and procedures for exchanging these between digital subscriber line (DSL) equipment, when the modes of operation of the equipment need to be automatically established and selected, but before signals are exchanged which are specific to a particular DSL Recommendation." G.994.1, §1. In other words, the transceivers execute the procedures defined in G.994.1 to agree on a common mode of operation (e.g., ADSL1, ADSL2, etc.) before they begin to execute one of the initialization protocols defined in the applicable transceiver standard.

1. T1.413 Issue 1 (1995)

155. The world's first ADSL standard, known as "T1.413 Issue 1," was released by ATIS in 1995. This section provides an overview of some of the features of T1.413 Issue 1 that are relevant to the Family 9 patents.

a. Initialization

156. T1.413 Issue 1 specifies an initialization procedure during which, among other things, the transmitter and receiver exchange configuration information. *See, e.g.*, ATIS ADSL Standard T1.413-1995, Network and Customer Installation Interfaces—Asymmetric Digital Subscriber Line (ADSL) Metallic Interface ("T1.413 Issue 1"), §12.

b. Frames and Superframes

157. T1.413 Issue 1 organizes the bits to be transmitted into *frames*. The transmitter transmits all of the bits in a frame in the same DMT symbol, with bits allocated to subcarriers in accordance with the bit allocation determined during the initialization procedure. To construct the DMT symbol, the bits in each frame are separated into groups of bits, each group being allocated to a different subcarrier. The number of bits in each group may be different.

158. T1.413 Issue 1 defines a collection of 69 consecutive frames as a *superframe*. I prepared Figure 7 below to illustrate the superframe structure. Each superframe is composed of 68 data frames, numbered 0 through 67, followed by a symbol known as the *synchronization symbol* (or "synch symbol," sometimes spelled as "sync symbol"). *See, e.g.*, T1.413 Issue 1, §6.2.1.1.

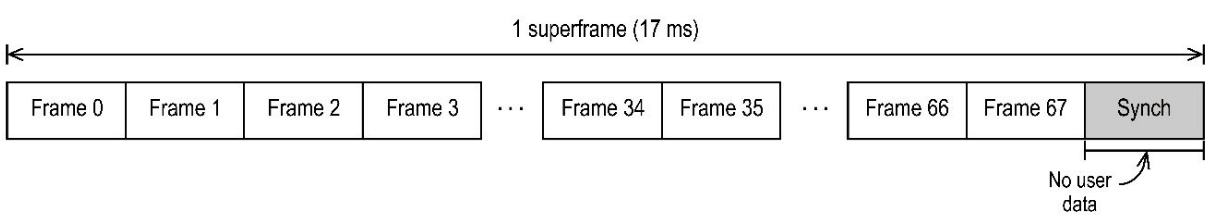


Figure 1: T1.413 Issue 1 superframe

159. Each data frame in a superframe carries data. In contrast, the synchronization symbol does not carry any data. Instead, it is a special symbol, known to the receiver, to help the receiver identify the superframe boundaries (and thus the frame boundaries). *See, e.g.*, *id.*, §§6.9.1.2, 6.9.3, 7.9.3.

c. <u>Latency Paths, FEC Coding, and Interleaving</u>

160. As explained above, it is well known that different types of data have different levels of sensitivity to delays that occur during transmission. T1.413 Issue 1 was designed to allow a single ADSL connection to transfer multiple types of data having different latency requirements (or delay tolerances). *See, e.g.*, ATIS Contribution T1E1.4/93-117, Revised FEC and Interleaving Recommendations for DMT ADSL ("T1E1.4/93-117"), Abstract ("We maintain the two-buffer parameterization explained in [3] that separates the data into delay-tolerant and delay-intolerant streams.").

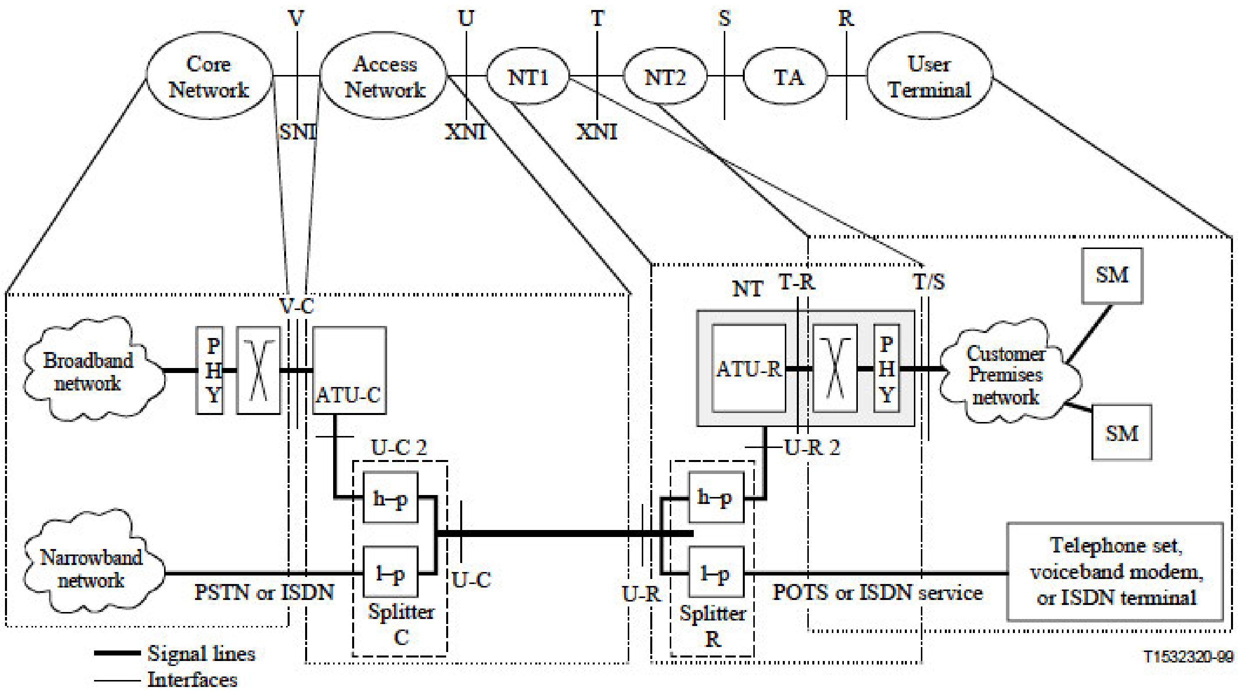
161. To allow the transmission of both delay-sensitive and delay-tolerant data over the same subscriber line, T1.413 Issue 1 supports *dual-latency* configurations using two independent *data paths*. Delay-sensitive data is routed to the so-called *fast path*, and delay-tolerant data is routed to the *interleaved path*. To mitigate errors in data transmission and reception, T1.413 Issue 1 specifies the use of FEC coding in the fast path, and both FEC coding and interleaving in the interleaved path. *See, e.g.*, T1.413 Issue 1, §6.4.1 (Reed-Solomon coding), §6.4.2 (interleaving). Thus, the key difference between the fast and interleaved paths is that data in the interleaved path is both FEC coded and interleaved, as described previously, to improve the effectiveness of the FEC code in the presence of impulse noise. In contrast, the fast path includes FEC coding, but does not include interleaving. Therefore, the fast path has lower latency than the interleaved path. *See, e.g.*, *id.*, §§6.4.1, 6.4.2.

2. T1.413 Issue 2 (1998) and ITU-T Recommendation G.992.1 (1999)

162. In 1998, ATIS published a revision of T1.413 Issue 1 ADSL, known as "T1.413 Issue 2." In 1999, the ITU released ITU-T Recommendation G.992.1, known as "ADSL1," much of which is identical to T1.413 Issue 2. Accordingly, I will describe aspects of T1.413 Issue 2 and G.992.1 together in this subsection. In general, I will refer to G.992.1; when necessary, I will point out where G.992.1 and T1.413 Issue 2 differ.

163. ITU-T Recommendation G.992.1 and T1.413 Issue 2 specify requirements for ADSL transceivers to enable high-speed data transmission between the network operator end of a subscriber line, where the ATU-C is located, and the customer end, where the ATU-R is located. ITU-T Recommendation G.992.1, Asymmetric digital subscriber line (ADSL) transceivers ("G.992.1"), i. The acronym "ATU" stands for "ADSL transceiver unit." *Id.*, §4.

164. Figure 1-1 of G.992.1, copied below, is an ADSL system reference model that illustrates the locations of the ATU-C and ATU-R within the network, as well as various defined interfaces between components.



NOTE 1 – The U-C and U-R interfaces are fully defined in this Recommendation. The V-C and T-R interfaces are defined only in terms of logical functions, not physical. The T/S interface is not defined here.
NOTE 2 – The V-C interface may consist of interface(s) to one or more (STM or ATM) switching systems.
NOTE 3 – Implementation of the V-C and T-R interfaces is optional when interfacing elements are integrated into a common element.
NOTE 4 – One or other of the high-pass filters, which are part of the splitters, may be integrated into the ATU-x; if so, then the U-C 2 and U-R 2 interfaces become the same as the U-C and U-R interfaces, respectively.
NOTE 5 – A digital carrier facility (e.g. SONET extension) may be interposed at the V-C.
NOTE 6 – Due to the asymmetry of the signals on the line, the transmitted signals shall be distinctly specified at the U-R and U-C reference points.
NOTE 7 – The nature of the customer installation distribution and customer premises network (e.g. bus or star, type of media) is for further study.
NOTE 8 – More than one type of T-R interface may be defined, and more than one type of T/S interface may be provided from an ADSL NT (e.g. NT1 or NT2 types of functionalities).
NOTE 9 – A future issue of this Recommendation may deal with customer installation distribution and home network requirements.
NOTE 10 – Specifications for the splitters are given in Annex E.

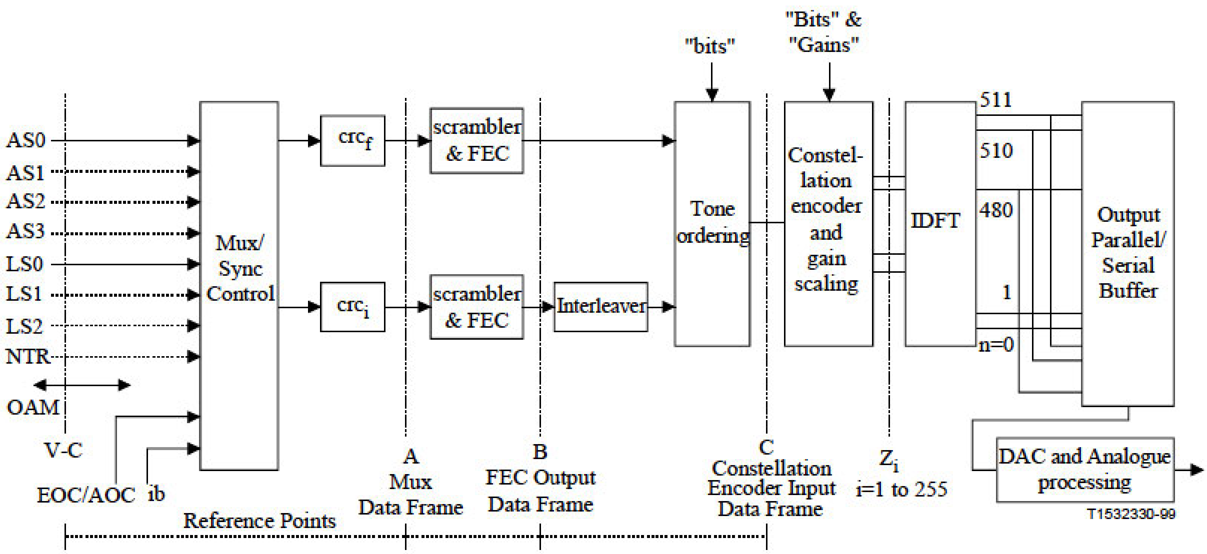
Figure 1-1/G.992.1 – ADSL system reference model

Id., Fig. 1-1.

165. Among other things, G.992.1 defines the modulation technique used in ADSL (namely, DMT), describes how data and overhead bits are organized into frames that are modulated for transmission, and it defines an initialization protocol. *Id.*, §§1, 7, 8, 10.

166. Figure 5-1 of G.992.1, copied below, is a block diagram of an ATU-C. *Id.*, §5.1.1; *see also id.*, Figure 5-2. The fast path is represented by the upper portion of the block diagram that includes the blocks "crc$_f$" and "scrambler & FEC," and the interleaved path is represented by

the lower portion of the block diagram that includes the blocks "crc_i," "scrambler & FEC," and "Interleaver." G.992.1 provides similar block diagrams for the ATU-R. *See, e.g.*, *id.*, Figs. 5-3, 5-4.



NOTE – Solid versus dashed lines are used to indicate required versus optional capabilities respectively. This figure is not intended to be complete in this respect, see clauses 6 and 7 for specific details.

Figure 5-1/G.992.1 – ATU-C transmitter reference model for STM transport

Id., Fig. 5-1.

167. Because of the addition of FEC redundancy bytes (in both the fast and interleaved paths) and interleaving (in the interleaved path), both of which are discussed further below, the data frames have a different appearance at different points along the data processing path. As shown above, Figure 5-1 of G.992.1 shows three reference points A, B, and C. The reference point A follows assembly of what is referred to as a *mux data frame*. The reference point B follows the mux data frames being scrambled and FEC encoded. The reference point C follows interleaving for the interleaved data, discussed below, and a process called tone ordering, which is not important for the discussion herein.

168. G.992.1 specifies the use of Reed-Solomon FEC coding for both fast and

interleaved data. *See, e.g.*, *Id.*, §7.6.1 ("R . . . redundant check bytes . . . shall be appended to K . . . message bytes . . . to form a Reed-Solomon codeword of size N = K + R bytes."). G.992.1 defines a parameter, S, which specifies the number of mux data frames per FEC codeword. *See, e.g.*, *id.*, §8.4.1.2 ("S = number of mux data frames per FEC codeword"). For the fast path, the value of S is always 1 (*see, e.g.*, *id.*, §§7.4.1.2.1, 7.6, Table 7-7), which means that each DMT symbol carries exactly one FEC codeword. In other words, in the fast path, the FEC codeword boundaries are identical to and coincident with the DMT symbol boundaries. For interleaved data, in addition to FEC coding, G.992.1 also requires the use of a convolutional interleaver with a programmable interleave depth. *Id.*, §7.6.3 ("The Reed-Solomon codewords in the interleave buffer shall be convolutionally interleaved. The interleaving depth varies, as explained in 7.4, but shall always be a power of 2.").

169. G.992.1 sets forth minimum required downstream and upstream FEC coding and interleaving capabilities for both the ATU-C and ATU-R, including all of the values of R, S, and D the transceivers must be capable of supporting. *See, e.g.*, *id.*, §§7.6, 8.6. Table 7-7, copied below, sets forth the minimum required downstream FEC coding and interleaving capabilities for the ATU-C transmitter (and the ATU-R receiver), and Table 8-3, also copied below, sets forth the corresponding minimum required upstream FEC coding and interleaving capabilities for the ATU-R transmitter (and ATU-C receiver).

Table 7-7/G.992.1 – Minimum FEC coding capabilities for ATU-C

Parameter	Fast buffer	Interleaved buffer
Parity bytes per R-S codeword	R_F = 0, 2, 4, 6, 8, 10, 12, 14, 16 (Note 1)	R_I = 0, 2, 4, 6, 8, 10, 12, 14, 16 (Notes 1 and 2)
DMT symbols per R-S codeword	S = 1	S = 1, 2, 4, 8, 16
Interleave depth	Not applicable	D = 1, 2, 4, 8, 16, 32, 64
NOTE 1 – R_F can be > 0 only if K_F > 0, and R_I can be > 0 only if K_I > 0.		
NOTE 2 – R_I shall be an integer multiple of S.		

The ATU-C shall also support upstream transmission with at least any combination of the FEC coding capabilities shown in Table 8-3.

Id., Table 7-7.

Table 8-3/G.992.1 – Minimum FEC coding capabilities for ATU-R

Parameter	Fast buffer	Interleaved buffer
Parity bytes per RS codeword	R_F = 0, 2, 4, 6, 8, 10, 12, 14, 16 (Note 1)	R_I = 0, 2, 4, 6, 8, 10, 12, 14, 16 (Notes 1 and 2)
DMT symbols per RS codeword	S = 1	S = 1, 2, 4, 8, 16
Interleave depth	not applicable	D = 1, 2, 4, 8
NOTE 1 – R_F can be > 0 only if K_F > 0 and R_I can be > 0 only if K_I > 0.		
NOTE 2 – R_I shall be an integer multiple of S.		

The ATU-R shall also support downstream transmission with at least any combination of the FEC coding capabilities shown in Table 7-7.

Id., Table 8-3.

170. G.992.1 specifies that both the ATU-C and ATU-R must support downstream transmission with any combination of the parameter values shown in Table 7-7, and they must support upstream transmission with any combination of the parameter values shown in Table 8-3. *See, e.g.*, *id.*, §§7.6, 8.6.

171. G.992.1 and T1.413 Issue 2 use the same superframe definition as T1.413 Issue 1, i.e., 68 data frames followed by a synchronization symbol that is identical to the synchronization symbol of T1.413 Issue 1. *See id.*, §§3.31, 7.4.1.1, 8.4.1.1. Figure 7-5 of G.992.1, copied below, illustrates superframe structure used by both the ATU-C and ATU-R.

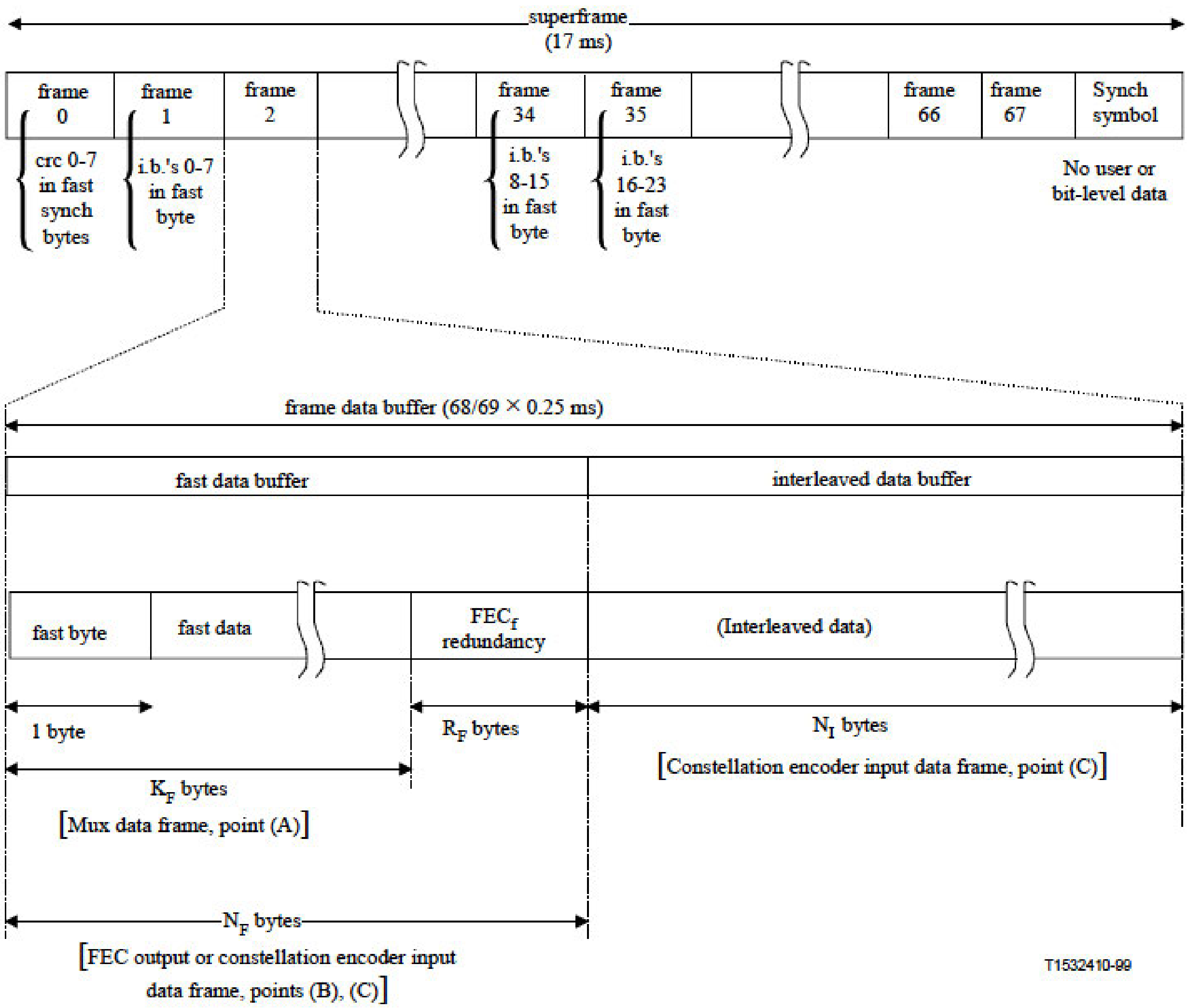


Figure 7-5/G.992.1 – ADSL superframe structure – ATU-C transmitter

Id., Fig. 7-5.

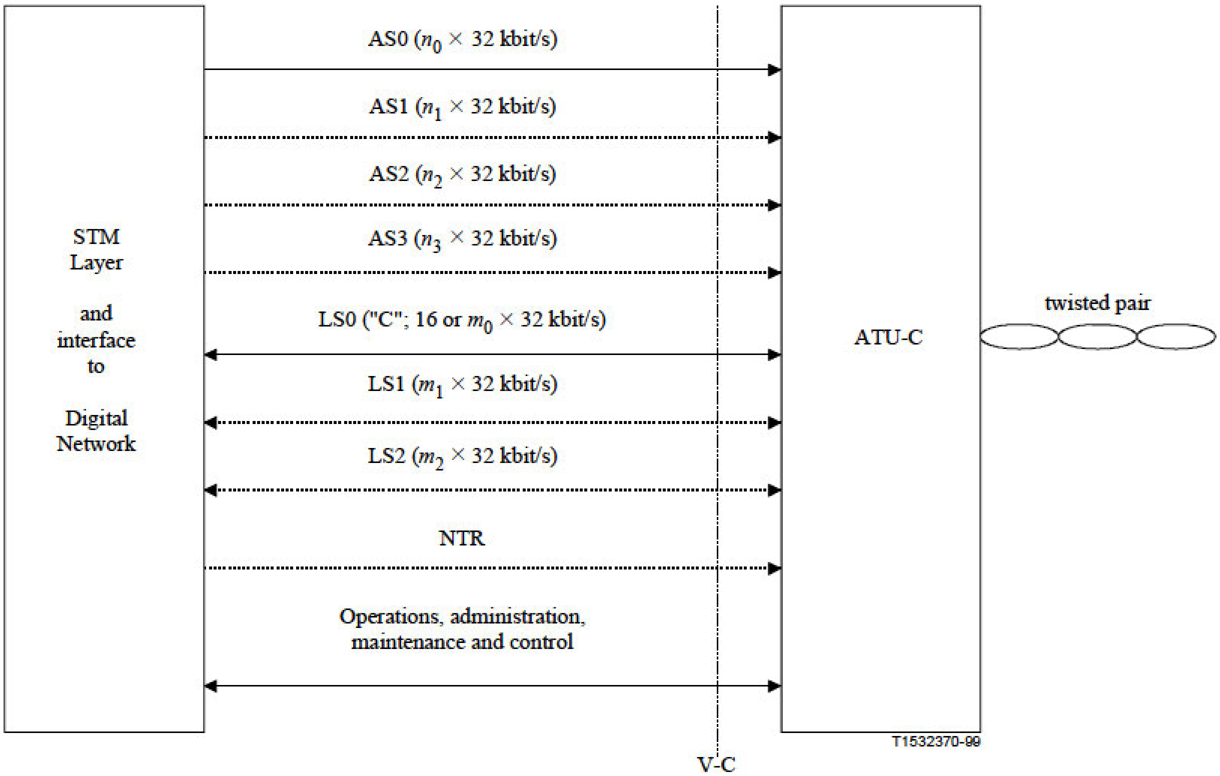
172. Like T1.413 Issue 1, to enable the transmission of both delay-sensitive and delay-tolerant data over the same subscriber line, T1.413 Issue 2 and G.992.1 support dual-latency configurations in which each data frame contains data from two data buffers, as shown in Figure 7-5 above. Delay-sensitive data is put into the fast data buffer for transmission, and delay-tolerant data is put into the interleaved data buffer. *Id.*, §7.4.1.2. But unlike T1.413 Issue 1, which always defines and uses both a fast buffer and an interleaved buffer, G.992.1 and T1.413 Issue 2 provide for the possibility that the user data requires only one latency path, whether interleaved or non-

interleaved. In this case, the amount of overhead can be reduced so that more of the total available bit rate of the connection is available for user data.

173. In the reduced overhead mode introduced in both T1.413 Issue 2 ADSL and G.992.1, the transmitter assigns data only to the fast data buffer or to the interleaved data buffer.

174. G.992.1 and T1.413 Issue 2 accommodate two ways that the two transceivers can transmit simultaneously over the subscriber line: frequency-division multiplexing (FDM) and echo canceling. *Id.*, §10.1.2 ("Manufacturers may choose to implement this Recommendation using either frequency-division multiplexing (FDM) or echo cancelling (overlapped spectrum) to separate upstream and downstream signals.").

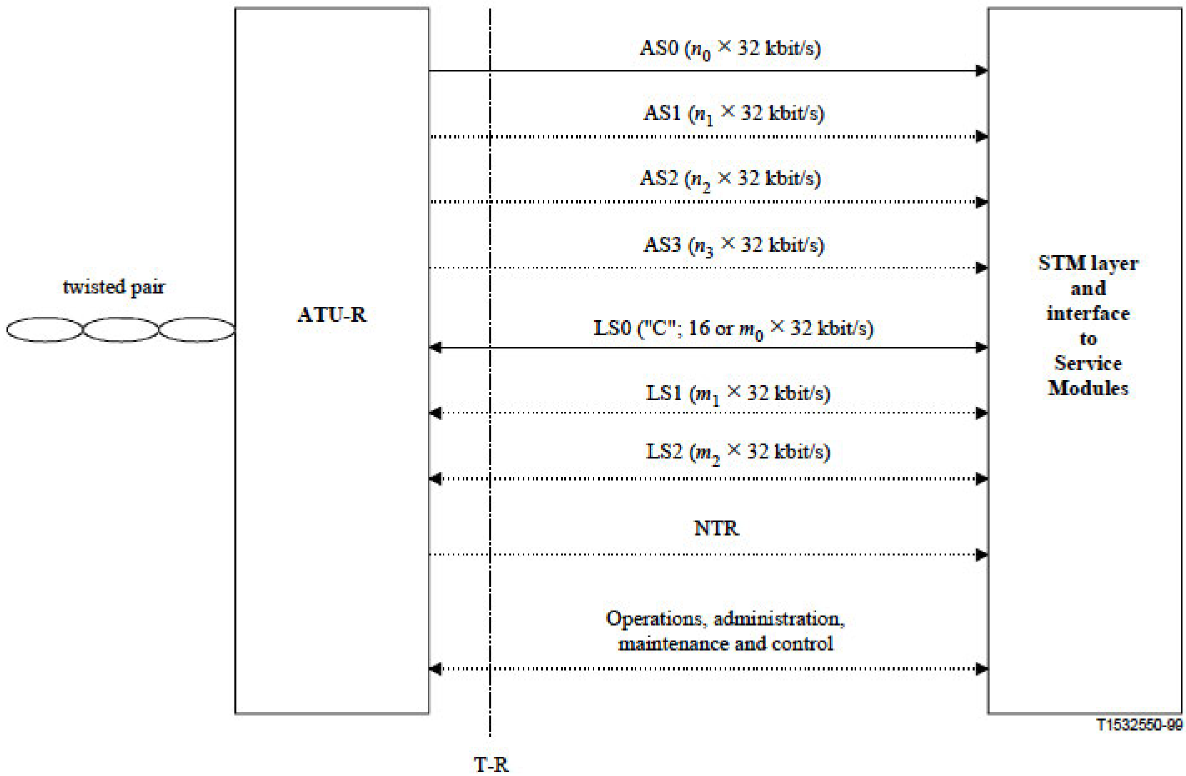
175. Figure 7-1 of G.992.1, copied below, shows the logical channels (also referred to as "bearer channels") defined for the ATU-C, and Figure 8-1 of G.992.1, also copied below, shows the logical channels defined for the ATU-R. *Id.*, §§7.1.1, 8.1.1.



NOTE – Optional bearer channels (both duplex and simplex) and features are shown with dotted lines.

Figure 7-1/G.992.1 – ATU-C functional interfaces for STM transport at the V-C reference point

Id., Fig. 7-1.



NOTE – Optional bearer channels (both duplex and simplex) and features are shown with dotted lines.

Figure 8-1/G.992.1 – ATU-R functional interfaces for STM transport at the T-R reference point

Id., Fig. 8-1.

176. As Figures 7-1 and 8-1 show, the ATU-C and ATU-R transmit and/or receive up to nine logical channels, all of which are communicated via a single twisted-pair line. To enable the different logical channels to share the same physical line, G.992.1 specifies rules the ATU-R and ATU-C follow to prepare data from all of the logical channels in use for transmission over the twisted pair. G.992.1 organizes transmitted and received data into data frames. *Id.*, §1. The ADSL transmitter transmits all of the bits in a data frame in the same DMT symbol. *Id.*, §§3.14, 7.4.1.1.

177. G.992.1 specifies initialization procedures that are "required in order for a physically connected ATU-R and ATU-C pair to establish a communications link." *Id.*, §10.1.1.

During initialization, "[i]n order to maximize the throughput and reliability of this link, ADSL transceivers shall determine certain relevant attributes of the connecting channel and establish transmission and processing characteristics suitable to that channel." *Id.* Each transceiver's receiver determines "the relevant attributes of the channel through defined transceiver training and channel analysis procedures." *Id.* The initialization procedure also allows the transceivers to establish "[c]ertain processing and transmission characteristics." *Id.* An exchange portion of initialization allows each receiver to communicate to the corresponding far-end transmitter certain transmission settings that it expects to see after the initialization procedure ends and the transceivers begin steady-state communication during Showtime. *Id.* For example, the ATU-C sends to the ATU-R four options for transport configuration for both upstream and downstream, each option including proposed values for the number of FEC redundancy bytes (R), the interleave depth (D), and the number of DMT symbols per FEC codeword (S). *Id.*, §10.8.3.

3. ITU-T Recommendation G.992.3 (2002)

178. In 2002, the ITU released ITU-T Recommendation G.992.3, known as "ADSL2" and, while in development, "G.dmt.bis." G.992.3 "describes the second generation of ADSL, based on the first generation ITU-T Rec. G.992.1." ITU-T Recommendation G.992.3, Asymmetric digital subscriber line transceivers 2 (ADSL2) ("G.992.3"), ii. Accordingly, G.992.3 builds on many of the aspects of G.992.1 to enable connections that support at least 8 Mbit/s downstream and at least 800 kbit/s upstream. G.992.3, §1. G.992.3 also adds a number of features and capabilities. *Id.*

179. Of note here, G.992.3 specifies the use of Reed-Solomon FEC and convolutional interleaving as two of the functions implemented by the physical-medium-specific transmission convergence (PMS-TC) function:

> [T]he ATU transmit PMS-TC function also provides procedures for: . . . insertion of redundancy for Reed-Solomon-based forward error correction; . . . and interleaving of data frames to spread the effect of impulsive impairments on the U interface. These functions are configured by a number of control parameters described in 7.5 to provide application-appropriate FEC protection, latency, and impulse noise immunity for each frame bearer. The values of the control parameters are set during initialization or reconfiguration of the ATU. The ATU receive PMS-TC function reverses each of the listed procedures so that the transported information may be recovered.

Id., §7.2.

> The FEC procedure inserts Reed-Solomon FEC redundancy octets to provide coding gain as an outer coding function to the PMD function. The FEC procedure of latency path function #p shall calculate R_p octets from $M_p \times K_p$ input octets. The octets are appended to the end of the input octets in the structure of FEC Output Data Frame at Reference Point B. When $R_p = 0$, no redundancy octets are appended and the values in the FEC Output Data Frame are identical to the input values. For all other values of R_p, the following encoding procedure shall be used to create the R_p octets: The FEC procedure shall take in M_p scrambled Mux Data Frames comprising message octets, m_0, m_1, … , $m_{M_p \times K_p - 2}$, $m_{M_p \times K_p - 1}$. The procedure shall produce R_p redundancy octets c_0, c_1, … , $c_{R_p - 2}$, $c_{R_p - 1}$. These two taken together comprise the FEC codeword of size $M_p \times K_p + R_p$ octets. The R_p redundancy octets shall be appended to the message octets to form the FEC Output Data Frame at Reference Point B. At the end of the initialization sequence, the FEC Function always starts with the first of M_p Mux Data Frames.

Id., §7.7.1.4.

> To spread the Reed-Solomon codeword and therefore reduce the probability of failure of the FEC in the presence of impulse noise, the FEC Output Data Frames shall be convolutionally interleaved. The interleaver creates the Interleaved FEC Output Data Frames at Reference point C, at the output of the latency path function. This procedure is followed by the frame multiplexing procedure. Convolutional interleaving is defined by the rule (using the currently defined values of the framing control parameters Dp and the derived parameter NFEC.p): Each of the NFEC.p octets B0, B1, … , BNFEC.(p-1) in an FEC Output Data Frame is delayed by an amount that varies linearly with the octet index. More precisely, octet Bi (with index i) is delayed by (Dp – 1)×i octets, where Dp is the interleaver depth.

Id., §7.7.1.5

180. Figure 7-6 of G.992.3, copied below, illustrates the PMS-TC functions of an ATU (either ATU-C or ATU-R), including the FEC coding and interleaving functions.

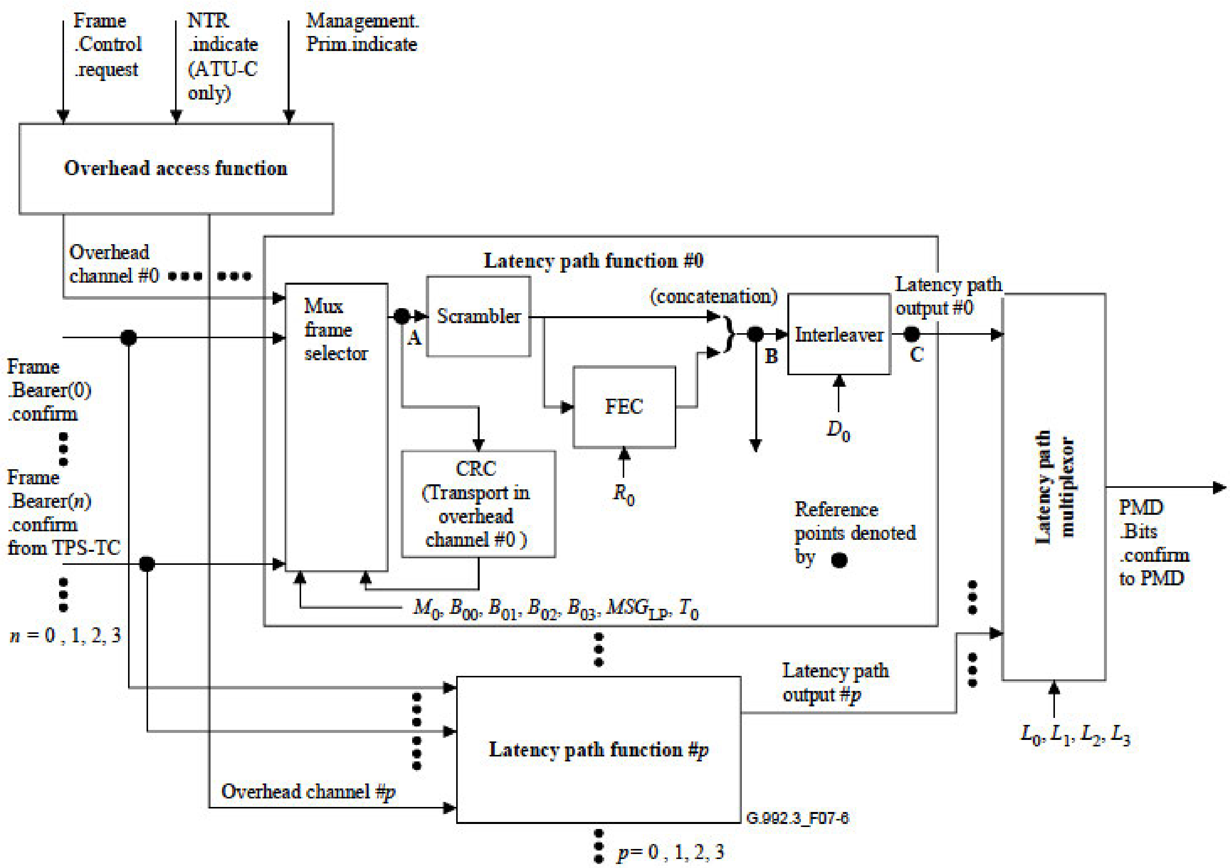


Figure 7-6/G.992.3 – Block diagram of transmit PMS-TC function

Id., Fig. 7-6.

181. Table 7-5, copied below, describes the data at the Reference Points A, B, and C in Figure 7-6.

Table 7-5/G.992.3 – PMS-TC function internal reference points

Reference point	Definition
A: Mux Data Frame	The data within a latency path function after the sync octet has been added.
B: FEC Data Frame	The data within a latency path function after the output of the FEC redundancy octets are merged with scrambled data.
C: Interleaved FEC Data Frame	The data and redundancy octets that have been interleaved. This is the output signal of a latency path function.

Id., Table 7-5.

182. A set of control parameters specified in G.992.3 controls the configuration of the PMS-TC function, including FEC coding and interleaving. *Id.*, §7.5. The control parameters include R_p, which is the number of Reed-Solomon (RS) redundancy octets per codeword (and per FEC Data Frame) in latency path function #p, and D_p, which is the interleaving depth in the latency path function #p. *Id.*

183. G.992.3 specifies valid values of each of the control parameters, including R_p and D_p. *Id.*, §7.6.2, Table 7-8. If $R_p = 0$ (FEC is off) then $D_p = 1$ (interleaving is off). *Id.*, §7.5.

184. G.992.3 specifies mandatory capabilities for the ATU-C and ATU-R for the mandatory latency path 0. *Id.*, §7.6.3.1. Specifically, ATU-Cs and ATU-Rs must support all valid values of both R_p and D_p for latency path 0 in both the downstream and upstream directions. *Id.*, §7.6.3.1, Table 7-9, Table 7-10. If the ATU-C and ATU-R support the optional latency paths, then during initialization the ATU-C and ATU-R identify the maximum values of R_p and D_p they can support on these optional latency paths, and all valid values of R_p and D_p up to and including these maximum values must be supported on the optional latency paths. *Id.*, §7.6.3.2, Table 7-11, Table 7-12.

4. <u>ITU-T Recommendation G.993.1 (2004)</u>

185. ITU-T Recommendation G.993.1, sometimes referred to as "VDSL1," specifies

aspects of very-high-speed digital subscriber lines (VDSL) to permit the transmission of asymmetric (downstream greater than upstream) and symmetric (downstream and upstream equal) aggregate data rates up to tens of Mbit/s on twisted pairs. G.993.1 provides for use of a wider bandwidth than is used in ADSL, namely, up to 12 MHz. ITU-T Recommendation G.993.1, Very high speed digital subscriber line transceivers (June 2004) ("G.993.1"), §1.

186. Among other things, G.993.1 specifies the use of FEC and interleaving in the transmitter. Figure 8-1 of G.993.1, copied below, illustrates the defined "slow" and "fast" paths. As shown, the "slow" path includes both FEC and interleaving, whereas the "fast" path includes FEC but not interleaving.

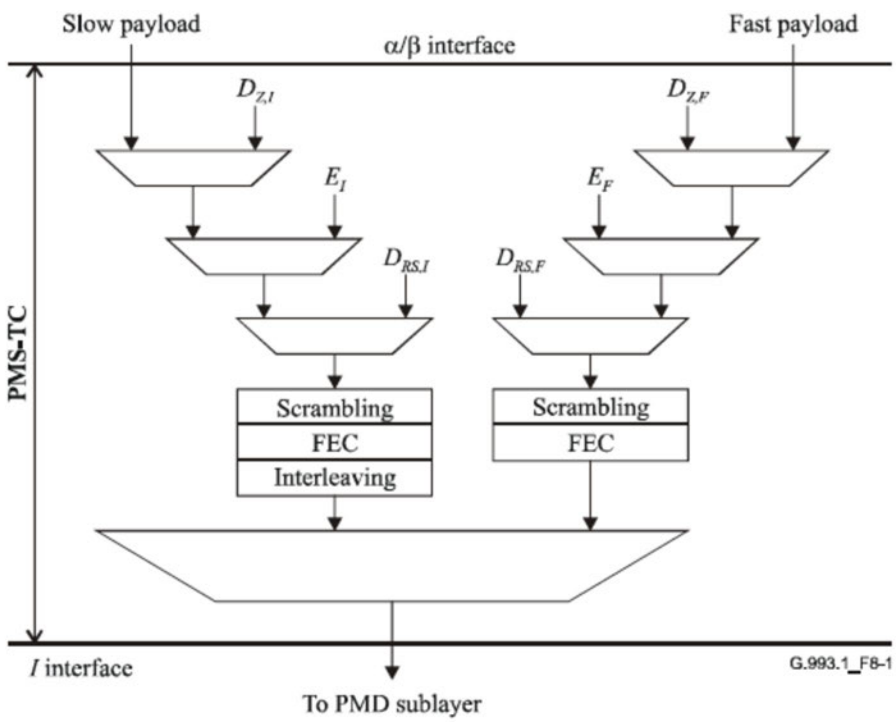


Figure 8-1/G.993.1 – Diagram of PMS-TC sublayer

G.993.1, Fig. 8-1.

187. G.993.1 specifies the use of "a standard byte-oriented Reed-Solomon code . . . to provide protection against random and burst errors" (G.993.1, §8.3) and interleaving "to protect

the data against bursts of errors by spreading the errors over a number of Reed-Solomon codewords." G.993.1, §8.4.1. Specifically, "the codewords shall be interleaved before transmission to increase the immunity of RS [Reed-Solomon] codewords to bursts of errors." G.993.1, §8.4.1. The interleave depth is programmable up to a maximum depth of 64 codewords when the codeword length is 255 bytes. *Id.* When the codeword is shorter than 255 bytes, the interleave depth can be larger than 64 codewords. *Id.*

188. G.993.1 specifies that "[i]t shall be possible to adjust the interleave depth via the management system to meet latency requirements." *Id.* The interleaver "uses a memory in which a block of I octets is written while an (interleaved) block of I octets is read." *Id.* G.993.1 teaches that the receiver requires the same size of memory for deinterleaving as the transmitter uses for interleaving. *Id.*

189. G.993.1 specifies an initialization procedure to allow the VTU-O and VTU-R to, among other things, exchange parameters such as Reed-Solomon settings and interleaver parameters. G.993.1, §12.4.1. Figure 12-7 of G.993.1, copied below, illustrates the timing of messages transmitted by the VTU-O and VTU-R during the channel analysis and exchange phase of VDSL1 initialization.

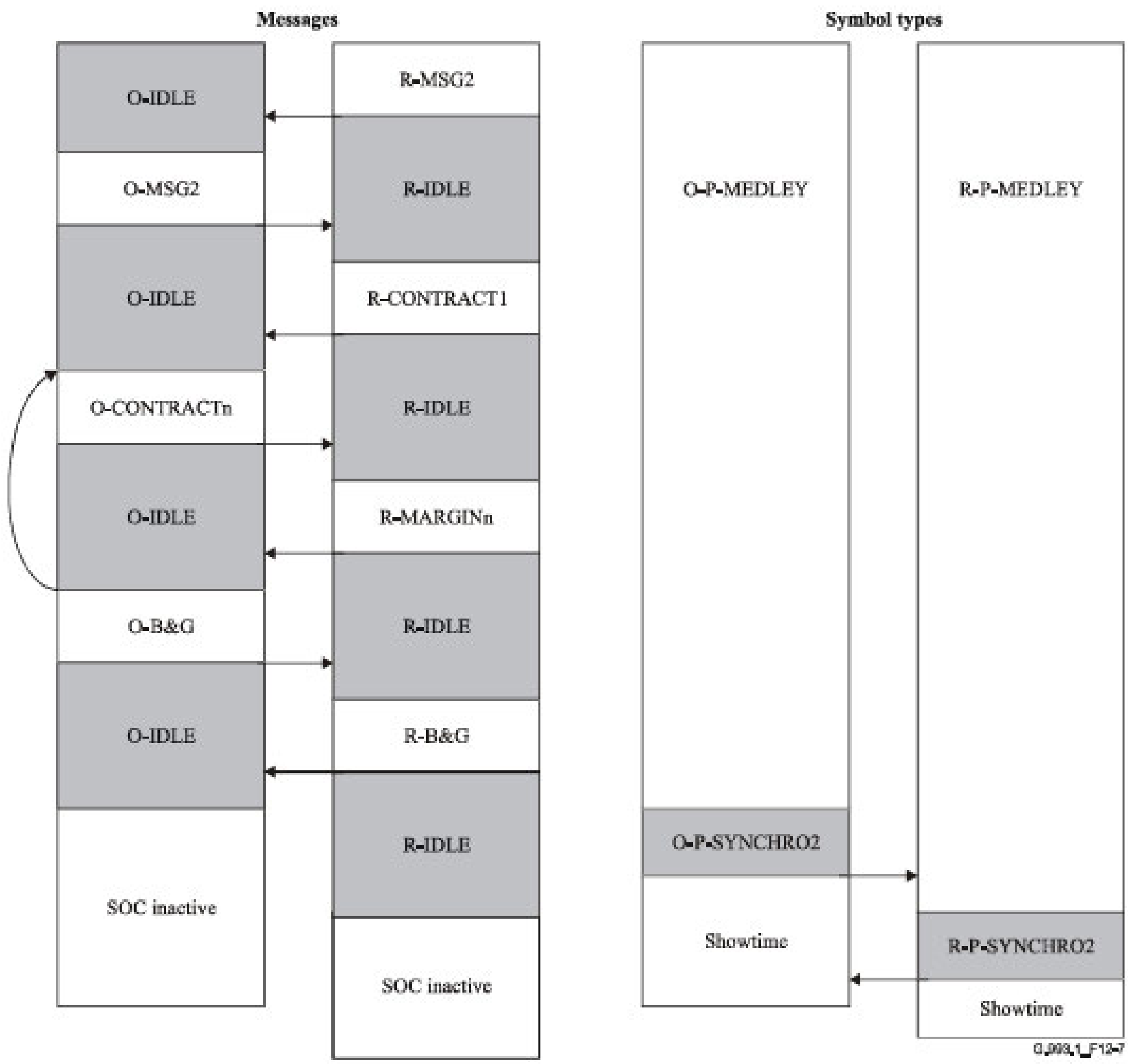


Figure 12-7/G.993.1 – Timeline of the channel analysis and exchange phase

Id., Fig. 12-7.

190. During the channel analysis and exchange phase, the VTU-R sends a message called R-MSG2, which transfers, among other things, the Reed-Solomon capabilities of the VTU-R (e.g., whether it can support only mandatory settings or all settings), the interleaver settings supported by the VTU-R (e.g., whether it can support only mandatory settings, all settings, or a number of additional settings), and the "maximal interleaver memory" in bytes. *Id.*, §12.4.6.1; §12.4.6.3.1.1.

191. After receiving R-MSG2, the VTU-O sends a message called O-MSG2, which transfers, among other things, the Reed-Solomon capabilities of the VTU-O (e.g., whether it can support only mandatory settings or all settings), the interleaver settings supported by the VTU-O (e.g., whether it can support only mandatory settings, all settings, or a number of additional settings), and the "maximum interleaver delay" in milliseconds. *Id.*, §§12.4.6.1, 12.4.6.2.1.1.

192. After receiving O-MSG2, the VTU-R sends a message called R-CONTRACT1, which contains "the proposed downstream contract based on the maximal number of bits in the slow channel based on the restrictions specified in O-MSG2 (i.e., as if only the slow channel will be used)." *Id.*, §12.4.6.3.1.2. The proposed downstream contract specifies, among other things, the bit rate for the slow channel (a multiple of 64 kbit/s), the Reed-Solomon settings for the slow channel (i.e., the codeword length and number of redundancy bytes), and the interleaver setting (i.e., the interleaver depth and block length). *Id.*, §12.4.6.3.1.2; 12.4.6.2.1.2.

193. After receiving R-CONTRACT1, the VTU-O sends a message called O-CONTRACTn, which contains a proposed upstream and downstream contract that is based on the capabilities of the VTU-O and VTU-R. G.993.1, §12.4.6.2.1.2. Table 12-25 of G.993.1, copied below, defines the O-CONTRACTn message.

Table 12-25/G.993.1 – Description of O-CONTRACTn

Field	Field or macro-field type
Message descriptor	Message code (see Table 12-12)
Downstream contract	Contract descriptor (see Table 12-26)
Upstream contract	Contract descriptor
EOC capacity (number of EOC bytes per frame)	1 byte
VOC capacity (value of *V*; see 8.5.5)	1 byte

Id.

194. The "Downstream contract" and "Upstream contract" in O-CONTRACTn are of

the type "Contract descriptor." *Id.* Table 12-26 of G.993.1, copied below, defines the contract descriptor, which includes information that allows the VTU-R to configure its interleaver (for upstream transmission) and deinterleaver (for downstream reception).

Table 12-26/G.993.1 – Contract descriptor

Field	Field or macro-field type	Remark
Rate in fast channel	2 bytes	In multiples of 64 kbit/s
RS setting in fast channel	2 bytes	B15 → B8: RS overhead B7 → B0: RS codeword length
Rate in slow channel	2 bytes	In multiples of 64 kbit/s
RS setting in slow channel	2 bytes	B15 → B8: RS overhead B7 → B0: RS codeword length
Interleaver setting	2 bytes	B15 → B8: *M* (Note) B7 → B0: *I*
NOTE – *I* must be a divider of the RS codeword length (in the slow channel).		

Id.

195. The downstream portion of the contract is based on the information provided by the VTU-R in R-CONTRACT1 and, ideally, is the same as the contract the VTU-R proposed in R-CONTRACT1. *Id.* As Table 12-26 indicates, both the upstream and downstream portions of the proposed contract include, among other things, the bit rate for the slow channel (a multiple of 64 kbit/s), the Reed-Solomon settings for the slow channel (i.e., the codeword length and number of redundancy bytes), and the interleaver setting (i.e., the interleaver depth and block length). *Id.*

196. G.993.1 also discloses the use of PTM-TC. For example, Annex H of G.993.1 is dedicated to PTM-TC. Furter, "the functional model of packetized data transport is presented in Figure H.1" below. *Id*. at Annex H.

5. <u>ITU-T Recommendation G.993.2 (2006)</u>

197. ITU-T Recommendation G.993.2, which is entitled "Very high speed digital subscriber line transceivers 2 (VDSL2)," "is an enhancement to ITU-T Rec. G.993.1 . . . that

supports transmission at a bidirectional net data rate (the sum of upstream and downstream rates) up to 200 Mbit/s on twisted pairs." ITU-T Recommendation G.993.2 (2006) ("G.993.2"), §1. G.993.2 specifies the use of DMT modulation and incorporates components from G.993.1 (VDSL1), G.992.3 (ADSL2), and G.992.5 (ADSL2 plus). *Id.*, §1.

198. G.993.2 provides much more flexibility than other DSL standards. For example, G.993.2 was developed to allow VDSL2 services to be deployed "from central offices, from fibre-fed cabinets located near the customer premises, or within buildings." G.993.2, §1. G.993.2 transceivers can "provide reliable high data rate operation on short loops" as well as "reliable operation on loops up to approximately 2500 metres of 26 AWG (0.4 mm)." *Id.* Data rates can be asymmetric (greater downstream than upstream) or symmetric (equal rates in both directions). *Id.*

199. Because G.993.2 "defines a wide range of settings for various parameters (such as bandwidth and transmitter power) that could potentially be supported by a transceiver," G.993.2 specifies "profiles" that allow VDSL2 transceivers to support subsets of all possible settings for various parameters (e.g., bandwidth, transmitter power, etc.). G.993.2, §1. The profiles "allow[] vendors to limit implementation complexity and develop implementations that target specific service requirements." *Id.*

200. A G.993.2-compliant transceiver must comply with at least one profile and can support multiple profiles. G.993.2, §6.1. G.993.2 defines eight profiles: 8a, 8b, 8c, 8d, 12a, 12b, 17a, and 30a. *Id.*

201. G.993.2 specifies the use of frequency-domain duplexing (FDD) to allow the subscriber line to be used at the same time for downstream and upstream transmission. G.993.2, §7.1. Frequency bands are allocated either for downstream transmission, for upstream transmission, or for neither. The allocation of specific frequency bands to the two directions of

transmission is referred to as the "band plan." *See, e.g.*, G.993.2, §7.1.

202. G.993.2 distinguishes between band plans for transceivers that use frequencies from near DC to 12 MHz, and band plans for transceivers that use frequencies from near DC to 30 MHz. G.993.2, §§7.1.1, 7.1.2. The maximum frequency that can be used by a transceiver to transmit or receive depends on the transceiver capabilities, the selected band plan, and the selected profile. *Id.*, §7.1.

203. The band plan for the frequency range up to 12 MHz is shown in Figure 7-1 of G.993.2, copied below:

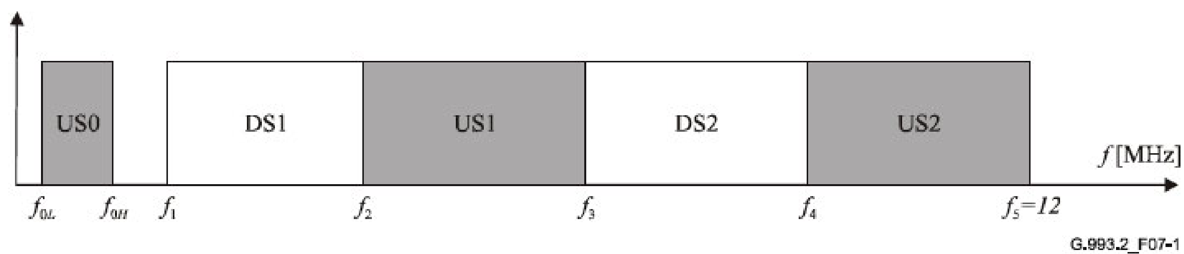


Figure 7-1/G.993.2 – Band plan in the frequency range up to 12 MHz

G.993.2, §7.1.1, Fig. 7-1.

204. The band plan shown in Figure 7-1 includes five bands: US0, DS1, US1, DS2, and US2. The bands beginning with the letters "US" are allocated for upstream transmission, and the bands beginning with "DS" are allocated for downstream transmission. G.993.2, §7.1.1.

205. The lowest-frequency band is denoted as US0. G.993.2, §7.1.1. Support of the US0 band is optional, which means a VTU-R can, but is not required to, be able to transmit in the US0 band, and, similarly, a VTU-O can, but is not required to, be able to receive in the US0 band. *Id.* As explained further below, the VTU-O and VTU-R convey their capabilities regarding usage of the US0 band during a handshake procedure conducted before the G.993.2 initialization procedure. *Id.*, §§12.3.2.1.1, 12.3.2.1.2, 12.3.2.2.1, 12.3.2.2.2. The G.993.2 initialization

procedure is discussed further below.

206. Figure 7-1 of G.993.2 illustrates that the US0 band resides between a lower frequency f_{0L} and an upper frequency f_{0H}. G.993.2, §7.1.1. The DS1 band resides between a lower frequency f_1, which is required to be higher than f_{0H}, and an upper frequency of f_2. *Id.* The US1 band resides between f_2 and an upper frequency f_3, the DS2 band resides between f_3 and an upper frequency f_4, and the US2 band resides between f_4 and 12 MHz. *Id.*

207. The values of the frequencies f_{0L}, f_{0H}, f_1, f_2, f_3, and f_4 are provided by region-specific annexes. G.993.2, §7.1.1. Annex A specifies the separation frequencies for use in North America, Annex B specifies the separation frequencies for use in Europe, and Annex C specifies the separation frequencies for use in Japan. *Id.*, Annex A, Annex B, Annex C. Annexes A, B, and C also specify whether and how transceivers can use frequencies between 12 MHz and 30 MHz. G.993.2, §7.1.2.

208. The band plan specified in Annex A for North America is copied below.

Figure A.1/G.993.2 – Band plan for North America

G.993.2, §A.1, Fig. A.1.

209. As indicated in Figure A.1, most of the separation frequencies in the Annex A band plan are fixed. For example, the US1 band spans the frequency range from 3.75 MHz to 5.2 MHz, the DS2 band spans the frequency range from 5.2 MHz to 8.5 MHz, and the US2 band spans the frequency range from 8.5 to 12 MHz. G.993.2, Fig. A.1.

210. The lowest-frequency bands US0 and DS1 have variable band-edge frequencies in

the Annex A band plan. G.993.2, §A.1. The value of f_{0L} can be as low as 4 kHz if there is no plain old telephone service (POTS) signal on the line or as high as 25 kHz (e.g., if there is a POTS signal sharing the line). *Id.* The value of f_{0H} can be any frequency from 138 kHz to 276 kHz but is required to be less than or equal to the value of f_1. *Id.*, §§7.1.1, A.1.

211. The value of f_1 is either 138 kHz or 276 kHz. G.993.2, §A.2.2, Table A.6.

212. Annex B of G.993.2 defines the band plans for use in Europe. G.993.2, §B.1. The band plans, denoted as "997" and "998," are specified in Table B.1, copied below:

Table B.1/G.993.2 – Band plans

<table>
<tr><th>Band plan</th><th colspan="7">Band-edge frequencies
(as defined in the generic band plan Figure 7-1)</th></tr>
<tr><th></th><th>f_{0L}
kHz</th><th>f_{0H}
kHz</th><th>f_1
kHz</th><th>f_2
kHz</th><th>f_3
kHz</th><th>f_4
kHz</th><th>f_5
kHz</th></tr>
<tr><td rowspan="2">997</td><td>25</td><td>138</td><td>138</td><td rowspan="2">3000</td><td rowspan="2">5100</td><td rowspan="2">7050</td><td rowspan="2">12000</td></tr>
<tr><td>25</td><td>276</td><td>276</td></tr>
<tr><td rowspan="4">998</td><td>25</td><td>138</td><td>138</td><td rowspan="4">3750</td><td rowspan="4">5200</td><td rowspan="4">8500</td><td rowspan="4">12000</td></tr>
<tr><td>25</td><td>276</td><td>276</td></tr>
<tr><td>120</td><td>276</td><td>276</td></tr>
<tr><td>N/A</td><td>N/A</td><td>138</td></tr>
<tr><td colspan="8">NOTE 1 – Flexibility in the bandwidth used for US0 is under study in ETSI TC-TM6.
NOTE 2 – N/A in the columns f_{0L} and f_{0H} designates a band plan variant that does not use US0.</td></tr>
</table>

G.993.2, §B.1.

213. Annex B states that "[t]wo variants are defined for band plan 997, and four for plan 998, to accommodate different underlying services (POTS and ISDN), and different US0 bandwidths." G.993.2, §B.1.

214. Annex C of G.993.2 specifies the band plan for use in Japan. G.993.2, §C.1. The band plan is shown in Figure C-1, copied below:

Figure C.1/G.993.2 – The band plan between 25 kHz and 30 MHz

G.993.2, §C.1, Fig. C.1.

215. Annex C states that "[t]he use of US0 is for further study." G.993.2, §C.1.

216. The transceivers specified in G.993.2 have many of the same functionalities as the transceivers specified in earlier DSL standards (e.g., ADSL1 and ADSL2). Figure 5-1 of G.993.2, shown below, is a diagram illustrating the VTU-O and VTU-R.

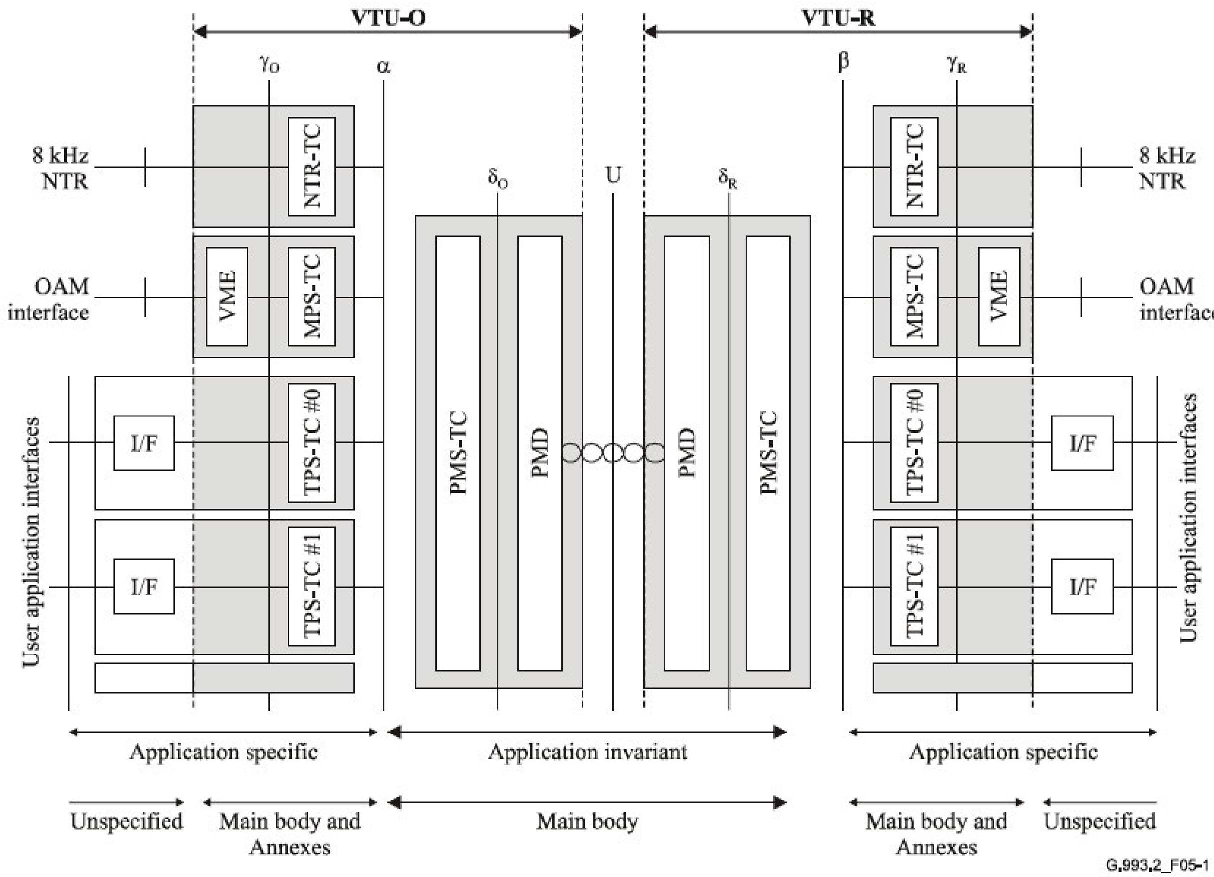


Figure 5-1/G.993.2 – VDSL2 and VTU functional model

G.993.2, Fig. 5-1.

217. Among other things, G.993.2 specifies the use of FEC and interleaving in the

PMS-TC layers of the VTU-O and VTU-R. G.993.2, §9.1. Figure 9-1 of G.993.2, copied below, illustrates PMS-TC layer, including the mandatory latency path, #0, and the optional latency path, #1, each of which includes both FEC and interleaving. *Id.*

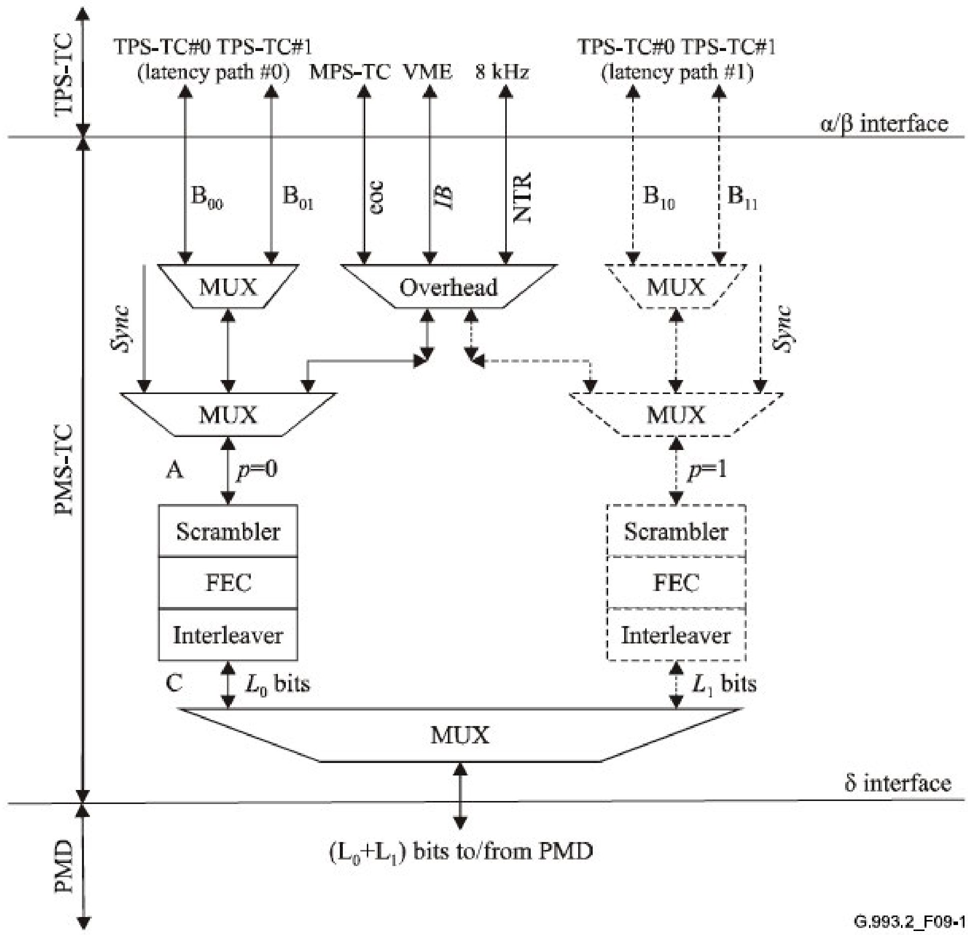


Figure 9-1/G.993.2 – PMS-TC functional model

G.993.2, Fig. 9-1.

218. Like other DSL standards, including G.993.1, G.993.2 specifies the use of "a standard byte-oriented Reed-Solomon code for forward error correction (FEC)," which "provides protection against random and burst errors." G.993.2, §9.3. Each Reed-Solomon code word contains $N_{FEC} = K + R$ bytes, where K is the number of data bytes, and R is the number of parity bytes. *Id.* In both the downstream and upstream directions, valid values of R are 0, 2, 4, 6, 8, 10,

12, 14, and 16 bytes. *Id.* The FEC codeword length, N_{FEC}, can be any integer number of bytes from 32 to 255, inclusive. *Id.* Each VTU-O and each VTU-R must support all valid values of R and N_{FEC}. *Id.*

219. G.993.2 also specifies the use of interleaving "to protect the data against bursts of errors by spreading the errors over a number of Reed-Solomon codewords." G.993.2, §9.4. The parameter I represents the interleaver block size in bytes, and the parameter D represents the interleaver depth. *Id.* In operation, the interleaver delays each of the I bytes in an interleaver block by an amount that varies linearly with the byte index, such that the byte B_j (with index j) is delayed by $\Delta[j] = (D - 1) \times j$ bytes. *Id.* The values of D and I are required to be co-prime, meaning they have no common divisor except 1. *Id.*

220. The overall (end-to-end) delay caused by the interleaver in the transmitter and the deinterleaver in the receiver is $(D - 1) \times (I - 1)$ bytes. G.993.2, §9.4. G.993.2 indicates that the interleaver in the transmitter needs at least $\frac{(D-1)\times(I-1)}{2}$ bytes of memory for interleaving, and the deinterleaver in the receiver on the other end of the line needs at least $\frac{(D-1)\times(I-1)}{2}$ bytes of memory for deinterleaving. *Id.*, §6.2.8.

221. G.993.2 restricts the length of the Reed-Solomon codeword, N_{FEC}, to an integer multiple of the interleaver block length I. G.993.2, §9.4. This restriction is given by the equation $N_{FEC} = q \times I$, where q is an integer between 1 and 8, inclusive. *Id.* VDSL transceivers are required to support all values of q from 1 to 8. *Id.*

222. The interleaver depth, D, is "set to meet the requirements for error-burst protection and latency." G.993.2, §9.4. The maximum value of the interleaver depth, denoted as D_{max}, is specified by the profile. *Id.*, §6.1. A VTU that supports a particular profile must support all integer values of D from 1 to the vaue of D_{max} specified for that profile. *Id.*, §9.4.

223. G.993.2 contemplates that a VTU-R might use shared memory to support its interleaving and deinterleaving needs. G.993.2, §12.3.5.2.1.3 (referring to "the portion of shared interleaver memory that the VTU-R can use to de-interleave the downstream data stream").

224. Unlike G.993.1, which uses bandwidth up to 12 MHz, (G.993.1, §1), G.993.2 allows the use of a much wider bandwidth: up to 30 MHz. G.993.2, §1.

225. To select a profile and prepare the transceivers for communication during Showtime, the VTU-O and VTU-R perform an initialization procedure. G.993.2, §12.1.3. Relative to G.993.1, G.993.2 provides "[i]mprovements to initialization." *Id.*, §1.

226. The G.993.2 initialization procedure includes a Channel Discovery phase during which the transceivers set power levels, choose a profile, and set other modulation parameters. G.993.2, §12.3.1. The phase that follows Channel Discovery is the Training phase, during which components such as equalizers and echo cancelers (if present) can be trained. *Id.*

227. The last phase of the initialization procedure is the Channel Analysis & Exchange phase, during which the transceivers exchange information about their capabilities, and they determine and exchange the parameter values they will use during Showtime. G.993.2, §12.3.1. Figure 12-8 of G.993.2, copied below, illustrates the stages of the Channel Analysis & Exchange phase of initialization.

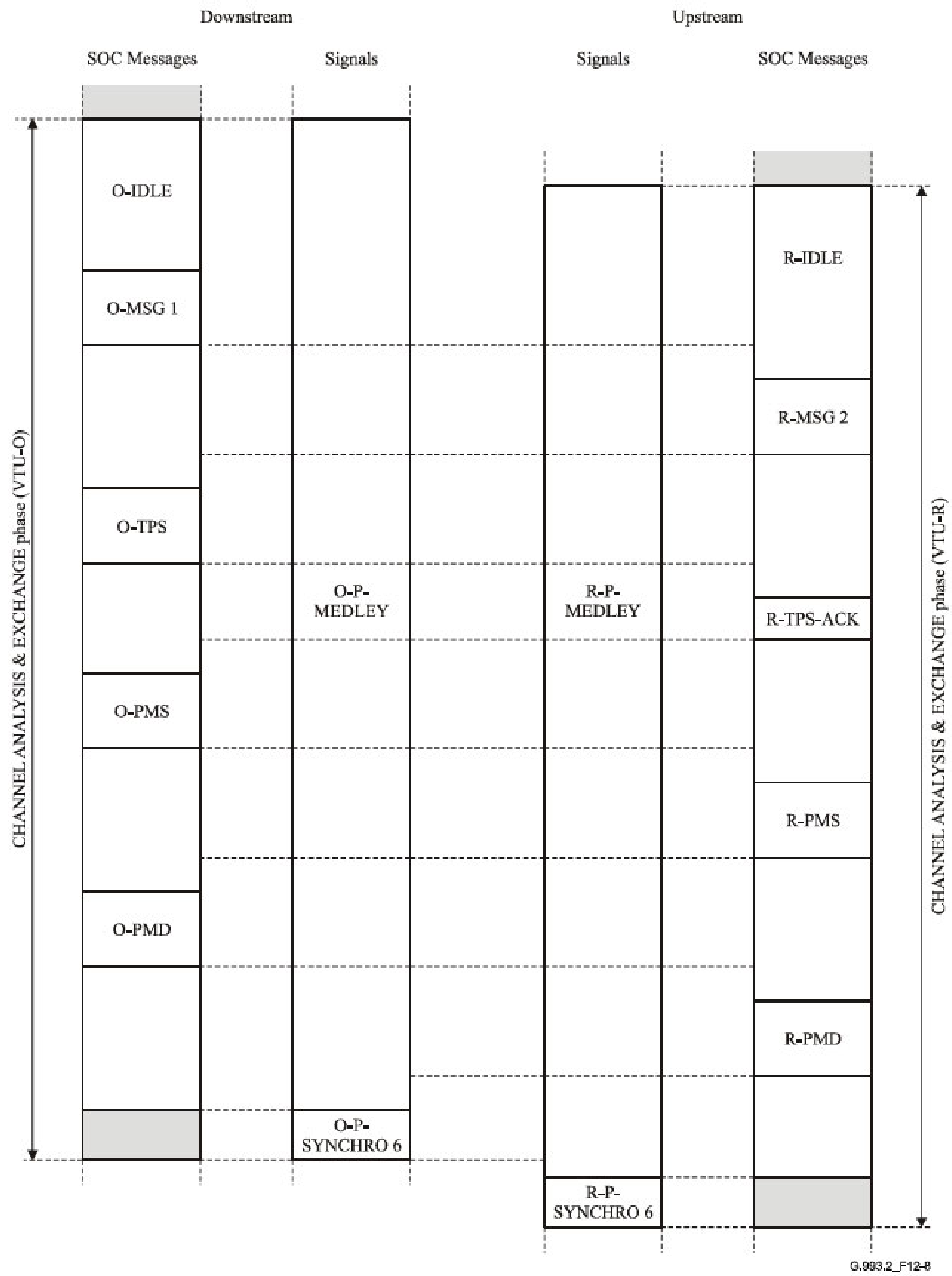


Figure 12-8/G.993.2 – Timing diagram for the stages of the channel analysis & exchange phase

228. In the O-MSG 1 message, the VTU-O sends its capabilities and various downstream configuration parameters to the VTU-R. G.993.2, §12.3.5.1. In response, the VTU-R sends the R-MSG 2 message to indicate its capabilities. *Id.* Next, the VTU-O specifies the

configuration of the bearer channels, both upstream and downstream, and their capabilities in the O-TPS message. *Id.* The VTU-R sends the R-TPS-ACK message to acknowledge the VTU-O's O-TPS message. *Id.*

229. The VTU-O then sends the O-PMS message to provide, to the VTU-R, the initial upstream PMS-TC parameters that will be used during Showtime. G.993.2, §12.3.5.1. These parameters include, for example, "the maximum interleaver delay that the VTU-R shall be allowed to use to de-interleave the data stream in downstream latency path #0," in bytes, and "the maximum interleaver delay that the VTU-R shall be allowed to use to de-interleave the data stream in downstream latency path #1" (if used), also in bytes. *Id.*, §12.3.5.2.1.3. The O-PMS message also provides values for the FEC coding and interleaving in the upstream direction, including the values of R, I, and D for latency path #0 and, if used, latency path #1. *Id.* G.993.2 states that the O-PMS message "specifies the portion of shared interleaver memory that the VTU-R can use to de-interleave the downstream data stream." *Id.*

230. The VTU-R then sends the R-PMS message to provide, to the VTU-O, the initial downstream PMS-TC parameters that will be used during Showtime. G.993.2, §12.3.5.1. These parameters include, for example, values for the FEC coding and interleaving in the downstream direction, including the values of R, I, and D for latency path #0 and, if used, latency path #1. *Id.*, §12.3.5.2.2.3. The VTU-R also indicates whether it will be using erasure decoding on any of the downstream latency paths. *Id.*

231. Next, the VTU-O sends the upstream bits per subcarrier and other information to the VTU-R in the O-PMD message, and the VTU-R responds with corresponding information for the downstream direction in the R-PMD message. G.993.2, §12.3.5.1. The VTU-O and VTU-R are then ready to transition to Showtime, which is coordinated using the O-P-SYNCRHO 6 and

R-P-SYNCRHO 6 signals. *Id.* The parameter settings negotiated and exchanged during the Channel Analysis & Exchange phase take effect starting from the first symbol of Showtime. *Id.*

232. To provide high robustness during some phases of initialization, G.993.2 specifies a retransmission protocol (the "RQ protocol"), "which allows the receiving VTU to ask for a retransmission of incorrectly received or missing message." G.993.2, §§12.3.3.1, 12.3.4.1. In contrast to the automatic repeat (AR) protocol, in which messages are automatically repeated, (G.993.2, §12.2.2.1), "[i]n RQ mode, each message . . . shall be sent only once." *Id.*, §12.2.2.2. The receiving VTU can request retransmission of the message by sending a request if the message has errors or has not arrived before a timeout timer expires. *Id.* The RQ mode is a stop-and-wait approach: "In RQ mode, a VTU shall never send a message (segment) prior to receiving an acknowledgement of the previously sent message (segment)." *Id.*

233. As did G.992.3 in the context of ADSL, G.993.2 recognizes that impulse noise can be problematic for VDSL and defines a parameter referred to as "impulse noise protection" or "INP." Specifically, G.993.2 states that "INP_p (impulse noise protection for latency path p) is defined as the number of consecutive DMT symbols or fractions thereof, as seen at the input to the de-interleaver, for which errors can be completely corrected by the error correcting code, regardless of the number of errors within the errored DMT symbols." G.993.2, §9.6.

234. The INP is a function of the FEC and interleaving parameters. G.993.2, §9.6. Specifically, the value of INP_p for a particular latency path *p*, in DMT symbols, is given by

$$INP_p = \frac{S_p \times D_p \times \left\lfloor \frac{R_p}{2 \times q_p} \right\rfloor}{N_{FEC,p}}$$

where, for the particular latency path *p*, S_p is the number of data symbols over which each FEC codeword spans, D_p is the interleaver depth, R_p is the number of redundancy bytes per FEC

codeword, q_p is the number of interleaver blocks in each FEC codeword, and $N_{FEC,p}$ is the FEC codeword size. *Id.*, §§9.6, 9.4, 9.5.4.

6. ITU-T Recommendation G.994.1 (2003)

235. ITU-T Recommendation G.994.1 is entitled "Handshake procedures for digital subscriber line (DSL) transceivers." G.994.1 specifies a way for DSL transceivers "to exchange capabilities and to select a common mode of operation," including "parameters relating to service and application requirements as well as parameters pertinent to various DSL transceivers." G.994.1, i. Specifically, G.994.1 "defines signals, messages and procedures for exchanging these between digital subscriber line (DSL) equipment, when the modes of operation of the equipment need to be automatically established and selected, but before signals are exchanged which are specific to a particular DSL Recommendation." *Id.*, §1.

236. As an example, as I explained above, during the handshake procedure that precedes G.993.2 initialization, the VTU-O and VTU-R exchange certain of their capabilities, including with regard to use of the optional US0 band. G.993.2, §§12.3.2.1.1, 12.3.2.1.2, 12.3.2.2.1, 12.3.2.2.2.

237. The procedures of G.994.1 are "an integral part of the start-up procedure for ITU-T Recs G.991.2, G.992.1, G.992.2, G.992.3, G.992.4 and G.992.5," with an expectation that "future DSL Recommendations will also be able to make use of this Recommendation." G.994.1, i.

238. The 2003 version of G.994.1 includes a "Retransmission Message" called "REQ-RTX" that allows either of the transceivers executing the handshake protocol (referred to in G.994.1 as "HSTUs") to request retransmission "in response to the reception of an errored frame." G.994.1, §7.15; *see also id.* at §10.5 ("Retransmission transactions occur whenever an HSTU receives an errored frame and wishes to transmit the REQ-RTX message to initiate a

retransmission instead of transmitting the NAK-EF [abort handshake] message."), §4 (indicating that HSTU stands for "Handshake Transceiver Unit"), §12 (indicating that NAK-EF message indicates that receiving transceiver is aborting handshake session).

239. In my opinion, as of the priority date of the Family 9 patents, a person having ordinary skill in the art would have been familiar with the relevant standards, including T1.413 Issue 1, T1.413 Issue 2, G.992.1, G.992.2, G.992.3, G.993.1, G.993.2, and G.994.1, as well as contributions to the working groups responsible for those standards, and other publications and technology related to DSL systems.

I. The Family 9 Patents

1. Overview of the Family 9 Patents

a. The '411 Patent

240. The '411 patent describes a communications system that uses retransmission. *Id.*, 1:25-27. Specifically, the '411 patent describes transceivers capable of using both retransmission and interleaving in conjunction with forward error correction to improve DSL communications in the presence of impulse noise.

241. The '411 patent notes that depending on channel conditions, one of retransmission or interleaving could be preferred over the other:

> Moreover, interleaving and RS coding methods and retransmission protocols provide different advantages with respect to error correction capabilities, latency, buffering requirements, and the like. For example, under certain configuration and noise conditions the interleaving/RS coding provides error correction/coding gain with less delay and overhead than the retransmission protocol (for packets that can be retransmitted). While under other conditions the retransmission protocol will provide better error correction with less delay and overhead than the interleaving/RS coding.

Id., 17:45-54.

242. The '411 patent describes using a shared memory for retransmission and

565. In summary, BI-089 in view of the general knowledge of a POSA discloses each and every element of claim 10 and renders claim 10 obvious as a result.

b. G.993.1 and BI-089 in view of the knowledge of a POSA

(1) Motivation to Combine G.993.1 and BI-089

566. As of April 12, 2006, a POSA would have been motivated to incorporate the ARQ techniques disclosed in BI-089 into G.993.1 VDSL transceivers to provide those transceivers with an additional tool to combat impulse noise, which has been known to be problematic for DSL systems since the mid-1990s. The POSA would have recognized that, as taught by BI-089, for some connections, and for some kinds of services, ARQ could be used instead of interleaving, in one or both transmission directions, to provide better impulse noise immunity. BI-089 at § 5; *see supra* § VII.D.3.c. Because it was known in the art that ARQ introduces delay jitter, which some types of services cannot tolerate, the person having ordinary skill in the art would have added ARQ to G.993.1 as an additional technique, but not as a wholesale replacement for interleaving. *See, e.g.*, PF-042, §1.1; BI-066 at 2. Instead, to provide the flexibility to choose the right tool to combat burst errors caused by impulse noise, a POSA would have made ARQ one of the tools the transceivers could choose to use to combat impulse noise.

567. G.993.1 was developed to provide reliable, high-speed transmission at rates up to tens of Mbit/s under a wide variety of conditions. For example, it could be deployed from a central office or from a cabinet near subscriber premises. It was designed to allow both symmetric and asymmetric bit rates to be provided to subscribers, and it was expected to be able to "overcome many types of ingress interference from radio and other transmission techniques that occur in the same frequencies" used by VDSL. G.993.1, §1.

568. G.993.1 recognizes that impulse noise is a significant concern for VDSL. Specifically, G.993.1 states that "deep interleaving will be required to provide adequate protection

DSL that has many commonalities with ADSL.

573. Furthermore, a POSA as of the priority date would have understood that, as taught by BI-089, ARQ is an alternative to interleaving, and that, from an information theoretic point of view, ARQ is superior to interleaving in the presence of burst errors.

574. Thus, a POSA would have been motivated, based on the disclosures of BI-089, to include ARQ in the VDSL systems of G.993.1 in order to improve their performance in the presence of burst errors and impulse noise.

575. It was well known in the art as of the priority date that impulse noise is frequency-dependent, and therefore affects some frequencies more than others. *See, e.g.*, W. Henkel and T. Kessler, "A Wideband Impulsive Noise Survey in the German Telephone Network: Statistical Description and Modeling," AEÜ, Vol. 48, 1994, No. 6, pp. 277-88 ("Henkel") (survey of impulse noise at frequencies below 5 MHz); C.F. Valenti and K. Kerpez, "Analysis of wideband noise measurements and implications for signal processing in ADSL environments," Proceedings of the IEEE International Conference on Communications, pp. 826 – 832, May 1994 ("Valenti") (survey of impulse noise at frequencies below 1 MHz); K.J. Kerpez and A.M Gottlieb, "The Error Performance of Digital Subscriber Lines in the Presence of Impulse Noise," IEEE Trans. On Comms., Vol. 43, No. 5, pp. 1902-05, May 1995 ("Kerpez") (survey of impulse noise at frequencies below 2 MHz). To my knowledge, as of the priority date, there was no survey data characterizing impulse noise at the higher frequency bands available to VDSL transceivers (e.g., up to 12 MHz). Therefore, as of the priority date, impulse noise remained a problematic impairment that was not well characterized in much of the bandwidth in which G.993.1 transceivers could operate (e.g., bandwidth up to 12 MHz). G.993.1, §1.

576. Therefore, a person having ordinary skill in the art would have been motivated to

provide, in G.993.1 VDSL transceivers, as much flexibility as possible and as many tools as possible to combat impulse noise. Based on the disclosures of BI-089, a person having ordinary skill in the art would have been motivated to add ARQ to the G.993.1 VDSL transceivers' suite of tools that could be used when appropriate to combat impulse noise.

577. It would have been straightforward for a person having ordinary skill in the art to add ARQ to G.993.1 VDSL transceivers. Adding ARQ to G.993.1 VDSL transceivers would merely have combined known methods that, in combination, would perform the same functions as they would separately. Although a POSA would likely have had to modify certain of the G.993.1 initialization messages to accommodate ARQ (e.g., to add fields that would indicate ARQ or interleaving, to indicate different parameter values, etc.), doing so would have been straightforward and well within the level of ordinary skill. A POSA would have been motivated to use the address field and/or control field in the G.993.1 PTM TPS-TC frame structure to indicate the order of packets being transmitted. Annex H.4.1.1

578. Moreover, allowing G.993.1 VDSL transceivers to use ARQ instead of interleaving in one or both directions of transmission would simply substitute one known method, ARQ, for another known method, interleaving, which is exactly what BI-089 suggests doing. BI-089 § 5. (noting that "[f]rom the information theoretic point of view, ARQ is the preferred method of burst error correction because of its ability to preserve and exploit the information in the noise correlation" and that "when ARQ is applied, interleaving [can] be reduced or eliminated thereby freeing memory resources for ARQ as well as to reduce or eliminate the interleaver delay so as to compensate for the delay introduced by ARQ"). ARQ was known in the art, and its effectiveness in mitigating burst errors was well known and is discussed at length in BI-089. *See id.*; *see also*, *e.g.*, Lin at 6-7; RFC 3366 at 7-8. The results of substituting ARQ for interleaving

in G.993.1 VDSL transceivers would have been utterly predictable: improved performance in the presence of impulse noise at a cost of increased delay jitter, which would be a desirable trade-off for certain applications (e.g. for non-streaming applications such as email, text, or web browsing, where delay jitter would not cause any problems).

(2) Independent Claim 9

(a) "An apparatus comprising:"

579. To the extent the preamble is limiting, this claim element is disclosed and rendered obvious by the combination of G.993.1 and BI-089.

580. As discussed above, BI-089 discloses an "apparatus" in the form of ADSL transceivers. *Supra* § VIII.B.3.a.(1).(a).

581. G.993.1 also discloses an "apparatus." Specifically, G.993.1 is entitled "Very high speed digital subscriber line transceivers," and discloses multicarrier VDSL transceivers, which a POSA would understand to be an "apparatus." G.993.1 at cover; § 8.5.1 ("A frame is a set of bytes carried by one DMT symbol."); § 9.2.2 ("All transceivers shall start transmission of DMT frames at the same time.").

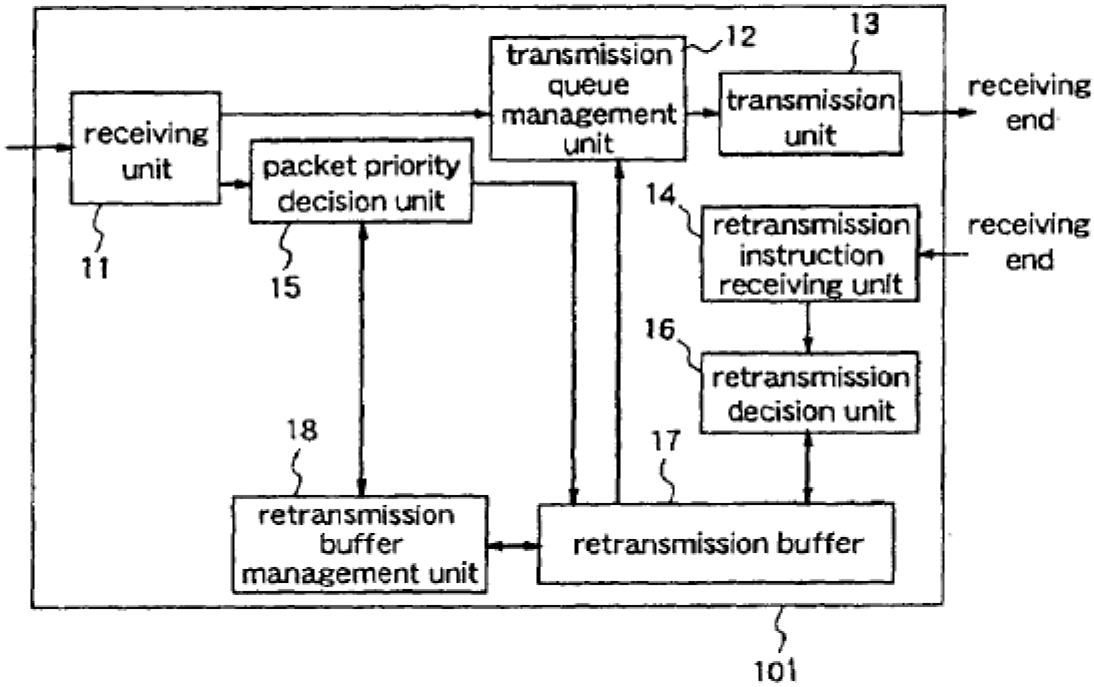
(b) "a multicarrier transceiver including a processor and memory operable to:"

582. The combination of G.993.1 and BI-089 discloses "a multicarrier transceiver including a processor and memory," and would render this limitation obvious to a POSA.

583. As discussed above, BI-089 discloses this limitation and/or renders it obvious. *Supra* § VIII.B.3.a.(1).(b).

584. G.993.1 also discloses "a multicarrier transceiver." *See, e.g.*, G.993.1, § 8.5.1 ("A frame is a set of bytes carried by one DMT symbol."); *id.* at § 9.2.2 ("All transceivers shall start transmission of DMT frames at the same time."). As would have been recognized by a POSA,

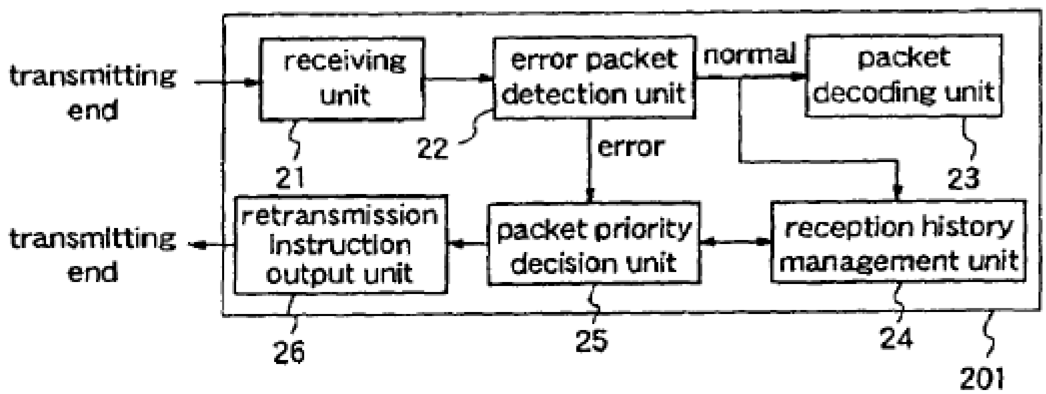
Fig.1 (a)



Id. Fig. 1(a).

629. Fukushima further describes a "data receiving apparatus," which is also capable of transmitting and receiving data. *See, e.g.*, Fukushima 15:51-16:5 ("data receiving apparatus" "includes a receiving unit" and a "retransmission instruction output unit" which "outputs a request for retransmitting…, toward the transmitting end"). Fukushima depicts an example of a "data receiving apparatus" in Figure 2:

Fig.2



Fukushima Fig. 2.

630. A POSA would have understood that the "data transmission apparatus" in Fukushima contained shared circuitry between the transmitter portion and receiver portion, and would have found this limitation obvious to the extent not explicitly disclosed. For instance, Fukushima describes the application of its invention to wireless WCDMA networks. Fukushima at 1:57-2:6. A POSA would understand that the practical implementation of wireless devices, such as WCDMA devices, involved the design of radio transceivers, incorporating shared circuitry between the transmitter and receiver sections of the devices. For instance, these shared circuits would typically include power supplies, clock and frequency generations circuits, user interfaces, and shielding, filtering, and other circuits. A POSA would also have found it obvious that the apparatus of Fukushima could be implemented within a DSL transceiver, which was understood to include shared circuitry as discussed above.

631. A POSA would have understood that the "data transmission apparatus" in Fukushima to include a processor, and would have found this limitation obvious to the extent not explicitly disclosed. In my experience, all DSL transceivers include a processor. For instance, a POSA would have known that conventional modems, such as those in ADSL as well as wireless

and analog modems, have long relied on digital signal processors. *See, e.g.*, Jacobsen, §11.6 (stating that "*DSL technology is becoming more and more based on software implementations on digital signal processors*" (emphasis added)); Ayre, §2 (disclosing different types of modems capable of digital signal processing). Fukushima refers to the use of standard modems, and it would have been obvious to implement Fukushima's apparatus within a modem, which would involve the use of a digital signal processor.

(c) <u>"receive a packet using a forward error correction decoder and a deinterleaver,"</u>

632. The combination of Fukushima and G.993.1 discloses a transceiver operable to "receive a packet using a forward error correction decoder and a deinterleaver," and would render this limitation obvious to a POSA.

633. As discussed above, G.993.1 discloses multicarrier transceivers operable to receive a packet using a forward error correction decoder and a deinterleaver. *Supra* § VIII.B.3.b.(2).(c).

634. Fukushima also discloses that its transceivers are operable to transmit and receive packets using FEC. Specifically, Fukushima discloses that, "at the transmitting end, each packet to be transmitted is given *error correction codes* for the additional information relating to its sequence number, priority, etc." *Id.* Furthermore, "[a]t the receiving end, the additional information is subjected to *error correction according to the error correction codes* and, thereafter, *a retransmission request for the error packet is made in accordance with the additional information*." (emphasis added). *See id.* 17:59-18:32. More specifically, each packet's additional information is "*required for real-time transmission of each packet data*." *See id.* 13:32-39 (emphasis added); *see also, e.g.*, Fukushima, 26:45-28:34 (indicating that at least two of CRC coding, FEC, and retransmission may be used together).